LAPPEENRANNAN TEKNILLINEN YLIOPISTO

TEKNISTALOUDELLINEN TIEDEKUNTA

TIETOTEKNIIKAN LAITOS

LAPPEENRANTA-MALLI ALUEELLISEN JULKISEN VERKON TOTEUTUSTAPANA

Diplomityön aihe on hyväksytty 19.5.2010.

Työn tarkastajat: Jari Porras, Jouni Ikonen

Työn ohjaaja: Jouni Ikonen

Lappeenrannassa 10.6.2010

Tomi Lapinlampi

Teknologiapuistonkatu 2C32

53850 LAPPEENRANTA

tomi.lapinlampi@lut.fi

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto

Teknistaloudellinen tiedekunta

Tietotekniikan laitos

Tomi Lapinlampi

Lappeenranta-malli alueellisen julkisen verkon toteutustapana

Diplomityö

2010

70 sivua, 9 kuvaa, 4 taulukkoa ja 5 liitettä

Tarkastajat: Jari Porras

Jouni Ikonen

Hakusanat: WLAN, 802.11b, alueellinen julkinen verkko, operaattoririippumattomuus

Työssä perehdytään tapoihin, joilla alueellinen julkinen verkko voidaan toteuttaa. Työn lähtökohtana on näkemys, että tietoyhteiskunnassa pääsy verkkoon on perusedellytys. Tällöin lähes kaikissa kodeissa tulisi olla mahdollisuus kytkeytyä ja olla jatkuvasti kytkeytyneenä tietoverkkoon.

Lappeenranta-malli määrittelee tavan toteuttaa alueellisen julkisen verkon peruspalvelut. Mallin erityispiirteenä on mahdollisuus ilmoitusten esittämiseen verkon käyttäjille. Työssä arvioidaan Lappeenranta-mallin sopivuutta alueellisen julkisen verkon toteutustavaksi ja mitataan mallin suorituskykyä.

Työn osana toteutetaan Lappeenranta-malliin kuuluva yhdysliikennepiste Lappeenrannan teknillisen yliopiston käyttöön.

ABSTRACT

Lappeenranta University of Technology

Faculty of Technology Management

Department of Information Technology

Tomi Lapinlampi

Implementing an operator neutral access network with the Lappeenranta model

Thesis for the Degree of Master of Science in Technology

2010

70 pages, 9 figures, 4 tables and 5 appendices.

Examiners: Jari Porras

Jouni Ikonen

Keywords: WLAN, 802.11b, public access network, operator neutral

This master's thesis describes ways to implement a regional public access network. It is assumed, that in modern information society the access to network is a fundamental element. Thus almost all homes should be able to connect and remain constantly connected to the Internet.

The Lappeenranta-model defines a way to implement the basic services needed in a regional public access network. The special feature of the Lappeenranta-model is it's capability to show announcements to the network users. This master's thesis analyses Lappeenranta-model's suitability for implementing an operator neutral access network and the performance of the model.

As for practice an access controller is implemented as defined by the Lappeenranta-model for the use of Lappeenranta university of technology.

SISÄLLYSLUETTELO

Kuvat

Taulukot

LYHENNELUETTELO

ADSL	Asynchronous Digital Subscriber Line
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
DSL	Digital Subscriber Loop
ETSI	European Telecommunications Standards Institute
FDDI	Fiber Distributed Data Interface
FLASH-OFDM	Fast Low-latency Access with Seamless Handoff, Orthogonal Frequency Division Multiplexing
GNU	Gnu's Not Unix
GPL	General Public License
GPRS	General Packet Radio Service
GSM	Global System for Mobile-Communications
HomePNA	Home Phoneline Networking Alliance
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HTTPS	Hypertext Transmission Protocol, Secure
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPV6	Internet Protocol version 6
IPX	Internetwork Packet Exchange
ISC	Internet Software Consortium
ISDN	Integrated Services Digital Network
ISM	Industrial, Scientific and Medical
ISO	International Standards Organisation
LAN	Local Area Network

MAC	Media Access Control
MSL	Maximum Segment Life
PAM	Pluggable AUthentication Modules
PDU	Protocol Data Unit
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
SCSI	Small Computer System Interface
SDH	Synchronous Digital Hierarchy
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
U-NII	Unlicensed National Information Infrastructure
VTT	Valtion Teknillinen Tutkimuskeskus
WAN	Wide Area Network
WEP	Wired Equivalent Privacy
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WLL	Wireless Local Loop
WWW	World Wide Web

Alkusanat

Tämä diplomityö on tehty Lappeenrannan teknillisen yliopiston tietotekniikan laitoksella vuosina 2003-2010.

Haluan esittää suuret kiitokseni työn valvojalle, Jari Porrakselle, sekä työtä ohjanneelle Jouni Ikoselle mahdollisuudesta työskennellä kiinnostavan aiheen parissa ja toteuttaa palvelu, josta on kuluneiden vuosien aikana ollut hyötyä lukuisille opiskelijoille ja yliopiston henkilökunnalle.

Lappeenrannassa 10.6.2010

Tomi Lapinlampi

1 JOHDANTO

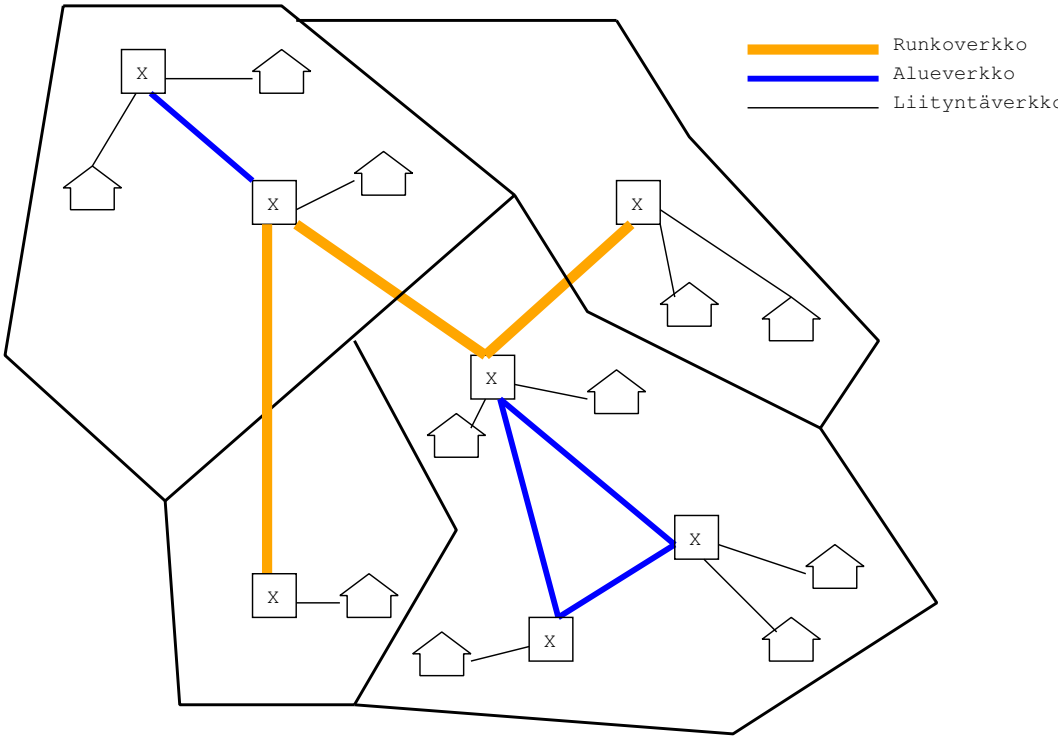
Langaton Lappeenranta -hankkeessa kehitetty Lappeenranta-malli määrittelee erään tavan toteuttaa alueellinen julkinen verkko. Tämän diplomityön tavoitteena on selvittää tapoja, joilla alueellinen julkinen verkko voidaan toteuttaa ja soveltaa näitä menetelmiä Lappeenranta-mallin jatkokehityksessä ja tuotteistuksessa. Lähtökohtana on näkemys, jonka mukaan tietoyhteiskunnassa pääsy verkkoon on perusedellytys. Lähes kaikissa kotitalouksissa tulisi olla mahdollisuus liittyä tietoverkkoon.

1.1 Alue- ja seutuverkot

Alueverkot ovat kuntien sisäisiä taajamia tai kaupunginosia yhdistäviä verkkoja, jotka on toteutettu runkoverkkojen tapaan valokaapeleilla ja radiolinkeillä. Alueverkkojen kanssa osin limittyvät seutuverkot tarjoavat yhteyksiä seutukunnan sisällä. Alue- ja seutuverkkojen tarkoituksena on sekä täydentää laajakaistaisten verkkoyhteyksien saatavuutta haja-asutusalueilla että tarjota perusta alueellisille verkkopalveluille. Kuva 1 havainnollistaa alue-, seutu- ja runkoverkkojen keskinäistä asemaa. Sekä alue- että seutuverkon omistajana voi olla tietoliikenneoperaattori, seutukunta tai kunnat yhdessä.

1.2 Alue- ja seutuverkkojen nykytila Suomessa

Liikenne- ja viestintäministeriö on tutkinut keväästä 2002 lähtien laajakaistapalveluiden saatavuutta käyttäjän näkökulmasta. Tutkimus, joka on julkaistu viisiosaisena raporttisarjana [7], antaa kattavan ja puolueettoman näkemyksen laaja-



Kuva 1: Alue-, seutu- ja runkoverkot

kaistan nykytilaan Suomessa.

Tutkimustuloksissa [6] arvioidaan, että noin 85% puhelinverkon keskus- ja keskitinalueista on liitetty valokaapelilla toisiinsa. Tästä voidaan edelleen laskea, että 98% väestöstä asuu valokaapelilla liitettyjen keskusten ja keskittimien piirissä, ja voisi siten teoriassa hankkia laajakaistayhteyden kotiinsa. Se ei kuitenkaan tarkoita laajakaistaistayhteyksien olevan lähes kaikkien saatavilla: taloudellisista ja teknisistä syistä tietoliikenneoperaattorit tarjoavat laajakaistaisia yhteyksiä vain noin 75-80%:lle väestöstä.

Yhteiskunnan tuki alue- ja seutuverkkojen rakentamisessa harvaan asutuille seuduille on välttämätöntä, sillä tietoliikenneoperaattorien liiketaloudellisin perustein rakentamat laajakaistayhteydet eivät johda riittävässä määrin yhteyksien tarjonnan parantumiseen. Yhteiskunnan suora tuki tietoliikenneoperaattoreille ja yhteiskunnan omien verkkojen rakentaminen voivat johtaa ongelmiin kaupallis-

ten toimijoiden ja julkisrahoitteisen hankkeiden välillä. Ongelmien välttämiseksi kaivataan yleistä toimintamallia, joka voisi perustua jo toteutuneissa hankkeissa saatuihin kokemuksiin. [5]

Verkottuminen ei voi olla itsetarkoitus, mutta se on kuitenkin tosiasia: Verkkoprojekteissa on kehitetty sekä perusrakenteita että verkkopalveluja asukkaiden käytettäväksi. Informaatioteknologiaan ja verkkopalvelujen kehittämiseen halutaan myös panostaa - ja pyrkiä sitä kautta kustannussäästöihin. [5] Tämä kuitenkin johtaa ennenpitkää palveluyhteiskunnasta itsepalveluyhteiskuntaan, jossa tietoteknisten valmiuksien, kuten henkilön omien kykyjen tai esimerkiksi laajakaistaisten yhteyksien puuttuminen voi lisätä väestöryhmien välistä eriarvoisuutta.

1.3 Alueverkkojen tekniikat

Alueverkkojen tekniikat voidaan jakaa kahteen pääryhmään: langallisiin ja langattomiin. Langallisten verkkojen toteutukset perustuvat joko kuparikaapeleihin eli pääosin olemassaolevaan kiinteään puhelinverkkoon tai valokaapeleihin. Langattomissa verkoissa käytetään yleensä langatonta lähiverkkoteknologiaa tai erityisiä kahden pisteen yhdistämiseen tarkoitettuja tekniikoita, kuten laajakaistaisia mikroaalto-, infrapuna- ja laserlinkkejä.

Langallisen verkon rakentamisessa suurimmat kustannukset syntyvät kaivettaessa kaapeleita maan sisään. [10] Langallinen, valokuidulla toteutettu verkko näyttää kuitenkin ainoalta keinolta toteuttaa siirtoverkko, jonka kapasiteetti ei lopu lähitulevaisuudessa kesken.

Langaton tekniikka on kehittynyt voimakkaasti erityisesti lähiverkkojen toteu-

tustapana. Teknisenä ratkaisuna se pystyykin kilpailemaan melko tasaväkisesti liityntäverkon toteutustapana langallisten tekniikoiden kanssa: useimmat verkon käyttäjät eivät vielä tänä päivänä vaadi siirtonopeuksia, jotka ovat saavutetavissa vain valokuidulla.

1.3.1 Langattomat tekniikat

Alueverkkojen langattomat tekniikat perustuvat yleensä IEEE:n 802.11-standardeihin. Lappeenranta-malli tarvitsee OSI-mallin tasolla 2 toimivan liityntäverkon, jonka 802.11-standardien mukaiset tekniikat langattomasti toteuttavat. Sen sijaan yleisiin televerkkoihin perustuvat GSM-, GPRS- ja UMTS-tekniikat eivät perusajatukseltaan sovellu alueverkkoihin - ne ovat yleensä valtakunnallisia, yhden teleoperaattorien tai teleoperaattorien yhteenliittymien hallitsemia suljettuja verkkoja.

Langattomista lähiverkkostandardeista vasta 802.11b saavutti suuren suosion, vaikka IEEE:n standardointityö langattomien lähiverkkojen parissa oli alkanut jo 1990-luvun alussa. IEEE 802.11b ja sen seuraajat, kuten 802.11a ja 802.11n tunnetaan myös nimellä WLAN. Ne ovat hajaspektritekniikkaa käyttäviä ISM (Industrial, Scientific and Medical) -taajuuksilla toimivia radioverkkoja. Näille verkoille on kansainvälisesti varattu taajuusalueet 902-982MHz, 2.4-2.4835GHz ja 5.725-5.85GHz. [16] Verkkojen ominaisuuksia on kuvattu taulukossa 1. ISM-taajuusalueilla liikennöinti on vapaata, eikä siihen tarvita viranomaisten lupaa. Paikalliset viranomaiset ovat tosin määritelleet tiettyjä rajoituksia lähetystehoon ja käytettävissä oleviin taajuusalueisiin liittyen. Suomessa käytössä ovat ETSI-standardin mukaisesti vain 2.4 ja 5.8 GHz:n taajuusalueet ja lähetysteho on rajattu 100 milliwattiin.

Taulukko 1: ISM-taajuuksilla toimivia radioverkkostandardeja

Standardi	802.11	802.11b	802.11a	802.11g	802.11n
Taajuusalue	2.4GHz	2.4GHz	U-NII[1]	2.4GHz	2.4GHz
Suurin nopeus	2Mbit/s	11Mbit/s	54Mbit/s	54Mbit/s	600Mbit/s

Tärkeimmät langattomat teknologiat ovat:

- IEEE 802.11

  IEEE 802.11 julkaistiin alunperin jo vuonna 1990, ja jatkokehityksen jälkeen vuonna 1997. Siinä määritellään 1 ja 2 Mbps signalointinopeudet taajuushyppely-, suorahajotus- ja infrapunalähetystekniikoilla. Radiotietä käytettäessä taajuusalueena on 2.4GHz.

- IEEE 802.11b

  Vuonna 1999 julkaistu IEEE 802.11b toimii 2.4GHz:n taajuusalueella. Lähetykseen voidaan käyttää taajuushyppely-, suorahajotus- tai infrapunalähetystekniikka. Standardin määrittelemillä uusilla 5.5Mbps ja 11Mbps signalointinopeuksilla on käytössä kuitenkin vain suorahajotustekniikka. Standardissa määriteltiin myös tiedon salausmahdollisuus 40-bittisellä RC4-salauksella. Tämä *WEP*-salaus osoittautui myöhemmin huonosti suunnitelluksi.

- IEEE 802.11a

  IEEE 802.11a toimii nk. U-NII-taajuusalueella, joka sijoittuu 5.15-5.85GHz välille. Suurin signalointinopeus on 54 Mbps. Osa kanavista on tarkoitettu sisäkäyttöön, osa ulkokäyttöön kahden pisteen välisille runkoyhteyksille.

- IEEE 802.11g

  IEEE 802.11g toimii 2.4GHz:n taajuusalueella. Suurin signalointinopeus on 54Mbps. 802.11g-standardin mukaiset laitteet ovat yleensä yhteensopivia

[1] U-NII-taajuusalue koostuu kolmesta kaistasta: 5.15 -5.25 GHz, 5.25-5.35 GHz ja 5.725-5.825 GHz. Euroopassa on käytössä näistä poiketen taajuusalue 5.470-5.725 GHz, josta on edelleen kansallisia poikkeuksia.

802.11b-standardin laitteiden kanssa, ja niitä voi käyttää yhdessä samassa verkossa. 802.11b-laitteiden liikennöidessä 802.11g-tukiaseman kanssa myös 802.11g-laitteiden liikenteen nopeus putoaa 802.11b:n tasolle.

- IEEE 802.11n

  IEEE 802.11n toimii 2.4GHz:n taajuusalueella. Suurin signalointinopeus on 600Mbps, joka saavutetaan uusien ominaisuuksien, kuten moniantennitekniikan ja 40MHz levyisten kanavien avulla. 802.11n pystyy jakamaan 2.4GHz taajuusalueen vanhempien standardien kanssa.

- WLL

  Wireless Local Loop on yleisnimi teknikoille, joilla puhelinverkon tilaajat yhdistetään langattomasti puhelinverkkoon (PSTN). Käsite kattaa mm. langattomat puhelinjärjestelmät ja kiinteät radiolinkit. [2]

- WiMAX

  Worldwide Interoperability for Microwave Access tarkoittaa IEEE 802.16:een perustuvaa verkkotekniikkaa, jonka mukaisten laitteiden yhteensopivuutta valvoo laitevalmistajista koostuva *WiMAX Forum*. WiMAX vastaa toimintaperiaatteeltaan perinteisiä langattomia lähiverkkoja, mutta tarjoaa pidemmän kantomatkan joka soveltuu esimerkiksi haja-asutusalueiden verkkoyhteyksien rakentamiseen.

- FLASH-OFDM

  FLASH-ODFM (Fast Low-latency Access with Seamless Handoff, Orthogonal Frequency Division Multiplexing) tarkoittaa laajakaistakäyttöön suuniteltua OFDM-modulointiin perustuvaa radioverkkoteknologiaa, jonka kehitti Flarion-niminen yritys. FLASH-ODFM ei perustu standardeihin, mutta se on silti saanut merkittävän aseman Suomen laajakaistamarkkinoilla Digitan rakennettua koko maan kattavan 450MHz taajuusalueella toimivan

FLASH-ODFM -verkon.[2]

1.3.2 Langalliset tekniikat

Merkittävimmät langalliset tekniikat ovat:

- Ethernet IEEE 802.3

  1970-luvulla kehitetty, 1985 standardoitu Ethernet on kehittynyt merkittävimmäksi lähiverkkojen teknologiaksi. Se valtaa jatkuvasti alaa muilta tekniikoilta, erityisesti alue- ja runkoverkoissa, vaikka SDH-pohjaiset järjestelmät säilyttänevät valta-asemansa vielä pitkään. [14]

- HomePNA

  HomePNA-tekniikalla voidaan toteuttaa talojen sisäisiä laajakaistaisia yhteyksiä olemassaolevan puhelinkaapeloinnin avulla normaalien lankapuhelinyhteyksien siitä kärsimättä. HomePNA sopii erityisesti taloyhtiöiden yhteisten laajakaistaratkaisujen toteutukseen siten, että esimerkiksi xDSL-pohjainen runkoyhteys kustannuksineen jaetaan useiden käyttäjien kesken.

- xDSL

  xDSL-tekniikat (esimerkiksi ADSL, VDSL ja SDSL) ovat tuoneet laajakaistaiset verkkoyhteydet taajama-alueilla lähes jokaisen ulottuville. xDSL on, ja tulee olemaan lähivuosien liityntäverkkojen merkittävin laajakaistateknologia.

[2]Kesäkuussa 2010 uutisoitiin Digitan luopuvan 450MHz taajusalueella toimivasta FLASH-ODFM -verkostaan asiakkaiden vähyyden vuoksi.

- POS/ATM -runkoverkkoteknologiat

  ATM, jonka tulevaisuus vielä muutama vuosi sitten näytti valoisalta, on väistymässä muiden, esimerkiksi POS- ja Ethernet-tekniikoiden tieltä. POS on ATM:n verrattuna merkittävästi tehokkaampi tapa käyttää runkoverkon kaistanleveyttä, sillä sen hyötykuorma verrattuna ATM:n on selvästi suurempi. [11] ATM-ratkaisut ovat myös tuntuvasti kalliimpia muut vertailukelpoiset teknologiat.

1.3.3 Lappeenranta-mallin teknologiavalinnat

Lappeenranta-mallin teknologiavalintoja ohjaa kaksi tekijää: Verkon tulee palvella käyttäjiä, ja siten olla yhteensopiva käyttäjien laitteiden kanssa. Toisaalta verkon rakentamisen kustannusten tulee pysyä kokonaisuuden kannalta järkevällä tasolla.

Teknologiavalintoja voidaan yksilöidä seuraavasti:

- Verkon käyttäjien laitekanta
  Käyttäjien laitteet noudattavat IEEE 802.11b-, IEEE 802.11g- ja IEEE802.11n -standardeja. Niinpä verkon laitteiden on tuettava näitä standardeja.

- Laitteiston hankinta- ja ylläpitokustannukset
  Laajasti valmistetut ja käytetyt, erityisesti Ethernet-verkkotuotteet ovat kustannustasoltaan edullisia. Näin ollen Lappeenranta-mallin runkoverkko on järkevintä toteuttaa Ethernet-tekniikalla.

2 OPERAATTORIRIIPPUMATTOMAT JULKISET VERKOT

Operaattoririippumattomalla julkisella verkolla tarkoitetaan sellaista IP (Internet Protocol)-verkkoa, johon liittyminen on vapaata ja jossa valinnaisia Internet-yhteyksiä tarjoaa yksi tai useampi Internet-operaattori. Verkon sisäisiä palveluita, kuten esimerkiksi viranomaisten tarjoamia palveluita on verkon toteutuksesta riippuen mahdollista käyttää myös ilman varsinaista Internet-yhteyttä. Yleensä operaattoririippumattomien julkisten verkkojen tarkoituksena on Internet-yhteyksien tarjoaminen, joskin alueellisten palvelujen yleistyessä tarve oikeisiin Internet-yhteyksiin saattaa vähetä tai osin jopa poistua. [13]

Operaattoririippumattomat julkiset verkot ovat uusi ilmiö Internetin yhteysvaihtoehtojen joukossa. Kaupallisten operaattorien rakentamien verkkojen rinnalle on noussut edullisella lähiverkkotekniikalla toteutettuja liityntäverkkoja, joiden taustalta löytyy tahoja joita ei perinteisesti ole pidetty tietoliikenneoperaattoreina: yliopistoja, yhdistyksiä, taloyhtiöitä ja yksityisiä henkilöitä. Verkkojen käyttäjät tarvitsevat kuitenkin yhteyttä Internetiin, jolloin kaupalliset operaattorit astuvat kuvaan. Muodostuu kokonaisuus jossa aiemmin omiin verkkoihinsa keskittyneet tahot joutuvat opettelemaan uudenlaisen yhteistoiminnan pelisääntöjä, jotta käyttäjille voitaisiin tarjota mahdollisimman hyvin toimivat verkoyhteydet.

2.1 Taustatekijät

Operaattoririippumattomien julkisten verkkojen synnyn taustalta löytyy monia tekijöitä, joista mikään ei olisi yksin voinut antaa alkusysäystä näiden verkkojen kehitykselle. Eräitä taustatekijöitä ovat:

- Televerkkojen vapautuminen

  Televerkkojen sääntelyn vähetessä on tullut mahdolliseksi rakentaa alueellisia verkkoja siten, että tarvittavat henkilö- ja taloudelliset resurssit ovat hyvin vähäiset verrattuna perinteiseen tietoliikenneoperaattorien toimintaan.

- Laitteiden kehitys

  Laajakaistayhteyksien luonnissa tarvittavien tietoliikennelaitteiden nopea halpeneminen sekä langattoman teknologian kehitys ovat osaltaan luoneet edellytyksiä operaattoririippumattomille julkisille verkoille.

- Yhteiskunnalliset tekijät

  Laajakaistaisten yhteyksien saatavuus ei ole itsestäänselvää kaikilla alueilla. Myös yhteyden kustannukset voivat olla liian suuret. Tämän vuoksi viranomaistahot useissa maissa ovat ryhtyneet toimenpiteisiin verkkoinfrastruktuurin rakentamiseksi joko kokonaan tai osittain yhteiskunnan varoin alueellisen tasa-arvon takaamiseksi. Suomessa laajakaistaisen yhteyksien saatavuutta käyttäjän näkökulmasta on selvitetty Liikenne- ja viestintäministeriön viisiosaisessa tutkimussarjassa, joka on päivitys vuonna 2000 ilmestyneeseen julkaisuun “Laajakaista kaikille? Tekniset ja taloudelliset edellytykset Suomessa (LVM:n julkaisu 41/2000)”. [7]

2.2 Tekniikka

Olemassaolevat operaattoririippumattomat julkiset verkot perustuvat yleensä lähiverkkotekniikoihin (IEEE 802.3, IEEE 802.11) näiden edullisuuden ja helpon hallittavuuden vuoksi. Vaikka perusta onkin yleensä sama, ovat kaikki käytännön verkkototeutukset varsin yksilöllisiä paikallisten vaatimusten ja verkon rakentajien mieltymysten mukaan.

Verkkojen ydin on yleensä toteutettu nopealla, joko Fast- tai Gigabit- Ethernet-tekniikalla. Tarpeen ja tarjonnan mukaan käytössä voi olla myös muihin tekniikoita, kuten puhelinverkkoon perustuvia DSL-ratkaisuja, ATM-yhteyksiä sekä langattomia linkkejä. Runkoyhteyksien luonnissa on mahdollisuuksien mukaan käytetty olemassaolevia yhteyksiä ja näin minimoitu kustannuksia.

Käyttäjät liittyvät verkkoihin yleensä joko langallisesti, jolloin liittymätapana on yleensä perinteinen puhelinmodeemi, ISDN tai laajakaistainen xDSL-, Ethernet- tai HomePNA-tekniikka. Liittymätapa voi olla myös langaton, jolloin tekniikkana on yleensä langaton lähiverkko. Usein verkoissa ovat käytössä molemmat menetelmät, koska kaikilla alueilla ei voida tarjota kiinteitä yhteyksiä, tai tietyille alueille halutaan tarjota langattomat yhteydet esimerkiksi vierailijoita varten.

2.3 Käyttäjien tunnistus operaattoririippumattomissa verkoissa

Operaattoririippumattoman verkon kannalta käyttäjien tunnistusta tarvitaan Internet-yhteyksien tarjoajien (ISP) valinnassa. Mikäli verkossa on useita yhteydentarjoajia, on erittäin tärkeää että käyttäjä voidaan yksilöidä riittävällä varmuudella ja ohjata hänen Internet-liikenteensä kulkemaan oikean yhteydentarjoajan kautta. Internetyhteyden tarjoajat puolestaan tarvitsevat käyttäjien tunnistusta laskutuksen ja käytön valvonnan työkaluksi. Lainsäädännössä edellytetään yhteydentarjoajien tallentavan ja säilyttävän lokitietoja jopa kuukausien ajan, joten tietojen oikeellisuus on pyrittävä varmistamaan mahdollisimman hyvin.

Sekä itse operaattoririippumatonta verkkoa hallinnoivan tahon että Internet-yhteydentarjoajien olisi helppoa ratkaista vahvan tunnistuksen ongelma vaatimalla verkon käyttäjiltä erityisen asiakasohjelmiston käyttöä. Tähän ei kuitenkaan ole yleensä haluttu mennä, sillä kustannukset on haluttu pitää mahdollisim-

man pieninä ja samalla maksimoida verkon käytettävyys erilaisilla laitteistoilla ja käyttöjärjestelmillä. Myös satunnaisia käyttäjiä palveltaessa, kuten esimerkiksi lentoasemille rakennetuissa verkoissa, on erillisten asiakasohjelmistojen vaatiminen usein hankalaa. Käyttäjien tunnistuksen ongelma on siis pitänyt ratkaista muilla keinoin, ja useat joko julkisia operaattoririippumattomia verkkoja rakentavat tai kaupallisia ratkaisuja valmistavat tahot ovatkin kehittäneet tarkoitukseen sopivia ohjelmistokomponentteja.

2.4 Tunnistusjärjestelmien vertailua

Operaattoririippumattoman verkon rakentajan täytyy siis joko valita jokin valmis malli autentikointijärjestelmän pohjaksi tai kehittää kokonaan oma järjestelmä. Eräät mallit ovat jo saavuttaneet suosiota alkuperäisen käyttökohteensa ulkopuolella, ja onkin perusteltua kysyä sopisivatko ne Lappeenranta-mallia paremmin operaattoririippumattoman verkon toteuttamiseen. Kysymykseen voidaan etsiä vastausta vertailemalla eri mallien ominaisuuksia.

Taulukko 2: Eräiden autentikointijärjestelmien ominaisuuksia

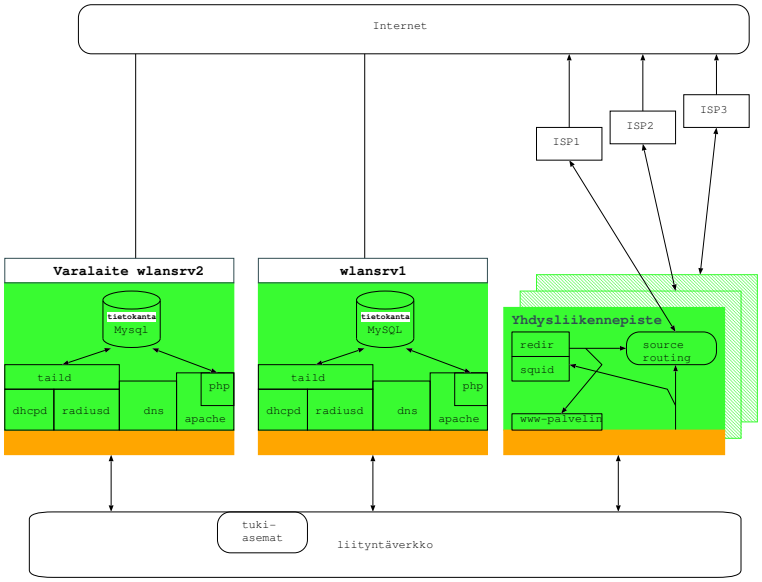
Nimi	WLPR.NET	StockholmOpen	NoCat	NetLogon
Lisenssi	ei tiedossa	GNU GPL	GNU GPL	GNU GPL
Yhteydentarjoajia	Useita	Useita	Yksi	Yksi
Alusta	Linux	Linux/FreeBSD	Linux/UN*X	Linux
Pääsyrajoitukset	IP-suodatus	IP/MAC -suodatus	IP/MAC- -suodatus	IP-suodatus
Autentikointi	MySQL	PAM-kirjasto	passwd, MySQL	SSH/POP3/ IMAP/NIS
Yhteyden purku	ajastettu	ARP ping, ICMP ping plugin, monitorointi	ajastettu	ARP ping, kytkimen SNMP-ohjaus
Osoiteavaruus	yksityinen	yksityinen/ julkinen	yksityinen/ julkinen	yksityinen/ julkinen

2.4.1 Lappeenranta-mallin ensimmäinen versio

Lappeenrannan teknillisen yliopiston tietoliikennetekniikan laboratorion WLPR.NET-projekti rakensi vuosina 2001-2002 Lappeenrannan alueelle koeverkon, jossa käytettiin Lappeenranta-mallin ensimmäisen version mukaista autentikointijärjestelmää. Lappeenranta-mallin ensimmäisen version rakenne on esitetty kuvassa 2. Käyttäjien autentikointi tapahtuu HTTPS-yhteyden kautta. Yhdysliikennepiste käyttää autentikointiin MySQL-tietokantaa, johon tiedot käyttäjistä on tallennettu. Verkon palomuuri suodattaa yhteyksiä vain IP-osoitteiden perusteella. Käyttäjien autentikoituessa tietyn yhteydentarjoajan käyttäjiksi heille annetaan uusi IP-osoite, joka kuuluu ennalta kyseiselle yhteydentarjoajalle määriteltyyn osoiteavaruuteen (esim. 10.1.0.0/16). Kaikki verkon käyttäjät kuuluvat silti saman aliverkon piiriin (esim. 10.0.0.0/8), ja voivat liikennöidä rajoituksitta toisten verkon käyttäjien kanssa. Verkon yhdysliikennepiste reitittää liikenteen Internetiin. [8] Yhdysliikennepisteen myöhempi versio tuki myös mahdollisuutta reitittää usean yhteydentarjoajan liikennettä pakettien lähdeosoitteen perusteella. [9] WLPR.NET:in erityispiirteenä voidaan pitää arkkitehtuuriin integroitua mahdollisuutta näyttää verkon käyttäjille mainoksia ja muita ilmoituksia. Mainoksia voidaan esittää esimerkiksi uuden käyttäjän aloittaessa verkon käyttö tai saapuessa jonkin tietyn tukiaseman alueelle. Mainokset esitetään käyttäjille läpinäkyvän www-välityspalvelimen avulla, jonka läpi kaikki porttiin 80 suuntautuva tcp-liikenne ohjataan. [9]

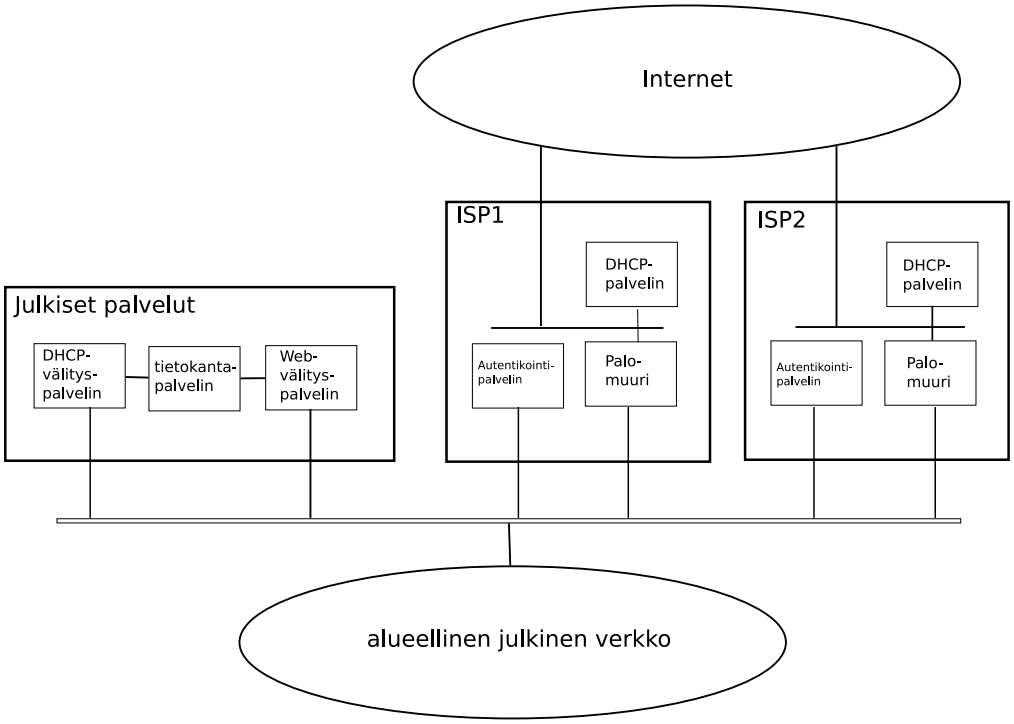
2.4.2 StockholmOpen

Kistan IT-yliopiston Open.Net-projekti, joka yhdistyi kesällä 2001 StockholmOpen-projektin kanssa on luonut Tukholman alueelle avoimen operaattoriippumattoman verkon, jossa toimii useita Internet-yhteydentarjoajia.



Kuva 2: Lappeenranta-mallin ensimmäinen versio (WLPR.NET)

Verkon rakenne on esitetty kuvassa 3. StockholmOpenin ainutlaatuinen piirre on erityisen välittäjäohjelmiston käyttö, jossa BOOTP-välittäjäagentti vastaanottaa ja välittää DHCP-pyynnöt eri yhteydentarjoajien DHCP-palvelimille sen mukaan, minkä yhteydentarjoajan asiakkaaksi kukin verkon käyttäjä on rekisteröitynyt. Näin on voitu kiertää yleisimmän ISC:n DHCP-palvelinohjelmiston rajoittuneesta hallintarajapinnasta aiheutuvat ongelmat. Internet-yhteydentarjoajien DHCP-palvelimet eivät vaadi muutoksia, mutta ne on sijoitettava verkkoteknisesti siten, ettei niistä ole suoraa yhteyttä itse julkiseen verkkoon. StockholmOpen-verkon autentikointipalvelin *Oasis* käyttää autentikointiin PAM-rajapintaa, johon voidaan liittää käytännössä rajattomasti erilaisia autentikointipalveluita. Autentikointipalvelin ohjaa *firewall control daemonin* avulla verkon reititintä, joka kykenee liikenteen suodatukseen sekä IP- että MAC-osoitteiden perusteella. Näin käyttäjien on hieman vaikeampaa, joskaan ei täysin mahdotonta anastaa toisen käyttäjän identiteettiä. [13]



Kuva 3: StockholmOpen-verkon rakenne

2.4.3 NoCat

NoCat on avoimen verkon luomiseen tähtäävä projekti Kaliforniassa, jonka tuotteena on syntynyt NoCatAuth. NoCatAuth on autentikointiohjelmisto, jonka avulla käyttäjät voidaan tunnistaa ja jakaa eri palveluluokkiin. NoCatAuth ei sovellu verkkoihin, joissa on useita Internet-yhteydentarjoajia, vaan sen kohdeympäristönä ovat yhteisölliset, yksittäisten Internet-yhteyksien jakamiseen perustuvat verkot. NoCatAuthin perustuu muiden autentikointimallien tapaan SSL-salattuun HTTP-yhteyteen, jonka kautta asiakas kirjautuu verkon käyttäjäksi saaden ennalta määritellyn palvelutason. Verkko on periaatteessa avoin kaikille käyttäjille, mutta sisäänkirjautumalla käyttäjä voi nostaa palveluluokkaansa. Korkeampaan palveluluokaan kuuluvan käyttäjän tietoliikenne on etuoikeutettua. Liikenteen luokittelun luotettavuutta parantaa verkon palomuurien suorittama sekä IP- että MAC-osoitteisiin perustuva suodatus. [12]

2.4.4 NetLogon

Linköpingin yliopiston UNIT-yksikön NetLogon-projekti on tuottanut NetLogon-ohjelmiston, jolla käyttäjät voidaan autentikoida ennenkuin heille suodaan oikeus verkon käyttöön. Järjestelmä koostuu tietokannasta, dhcp-palvelinohjelmistosta, WWW-palvelimesta, nimipalvelimesta sekä skripteistä, jotka on kaikki integroitu samaan laitteeseen. Laitealustana toimii RedHat-Linuxilla varustettu PC.

NetLogon toimii siten, että uusi käyttäjä ohjataan palomuurin REDIRECT-säännöllä sisäänkirjautumissivulle, ja kirjautumisen onnistuessa hänen IP-osoitteelleen luodaan sääntö joka sallii liikenteen ulkomaailmaan. NetLogon ei tue MAC-osoitteiden perusteella tapahtuvaa pakettien suodatusta, sillä se perustuu toistaiseksi Linuxin ipchains-työkaluihin joissa ei tätä ominaisuutta ole. NetLogonin kaltaisia järjestelmiä on kehitteillä ja käytössä myös muissa ruotsalaisissa yliopistoissa.

3 LAPPEENRANTA-MALLIN JATKOKEHITYS

Lappeenranta-malli määrittelee arkkitehtuurin, jonka avulla voidaan rakentaa operaattoririippumaton julkinen verkko. Verkon tärkeimpiä ominaisuuksia ovat operaattoririippumattomuus, usean palveluntarjoajan yhtäaikainen toiminta verkossa, avoin arkkitehtuuri sekä mahdollisuus esittää verkon käyttäjille tiedotteita ja mainoksia heidän fyysiseen sijaintiinsa perustuen. [9] [8] Lappeenranta-mallin ensimmäinen versio määriteltiin, tai se määrittyi useiden opinnäytetöiden ja todellisen verkon toteuksen myötä vuosina 2000-2002. Mallin jatkokehityksen tavoitteena on tuotteistus siten, että uudelleenkäyttö muissa sovelluskohteissa on mahdollista.

3.1 Lappeenranta-mallin toteutusvaihtoehdot

Hajautettu malli

Hajautetussa mallissa Lappeenranta-mallin toiminnalliset komponentit on hajautettu useisiin tietokoneisiin. Mallin etuja ovat yksittäisten tietokoneiden yksinkertaisempi konfigurointi, kriittisten palveluiden helpompi hajautus, monioperaattoriverkon helpompi toteutus ja tietoturvan helpompi hallittavuus. Mallin haittoja ovat suuret laitteistokustannukset, suuri fyysisen tilan tarve sekä, haluttaessa toimintavarmuutta esimerkiksi huoltosopimusten kautta, suuret käyttökustannukset.

Osittain hajautettu malli

Osittain hajautetussa mallissa Lappeenranta-mallin toiminnalliset komponentit on hajautettu useisiin tietokoneisiin, mutta kuitenkin niin, että yksittäinen tietokone sisältää useita järjestelmän komponentteja. Lappeenranta-mallin komponentit voidaan hajauttaa usealla eri tavalla. Valitsemalla käyttötarkoitukseen sopivin tapa, voidaan poimia parhaat puolet sekä täysin hajautetusta että keskitetystä ratkaisusta ja saavuttaa näin kustannussäästöjä.

Keskitetty ratkaisu

Keskitetyssä mallissa Lappeenranta-mallin toiminnalliset komponentit on koottu yhteen tietokoneeseen. Mallin etuja ovat vähäinen fyysisen tilan tarve ja edulliset laitteistokustannukset. Mallin haittapuolia ovat konfiguroinnin monimutkaisuus, eri ohjelmistokomponenttien vaikutus toisiinsa kuormitustilanteessa, tietoturvan mahdolliset puutteet sekä järjestelmän vaikea kahdennus. Myös monioperaattoriympäristön rakentaminen on vaikeaa.

3.2 Toteutus

Esimerkkitoteutustavaksi ja samalla WLPR.NET-verkon toteutustavaksi valittiin osittain hajautettu malli, jossa verkon päätoiminnallisuudet on hajautettu neljään pääosaan laitteisto- ja ylläpitoresurssien puitteissa. Malli on esitetty kuvassa 4.

- Yhdysliikennepisteet

Verkon yhdysliikennepisteet palvelevat kukin joko yhtä tai useampaa Internet-yhteydentarjoajaa. Käyttäjien tunnistusjärjestelmä on räätälöity kunkin yhteydentarjoajan tarpeisiin. Yhdysliikennepisteet on hajautettu verkon eri puolille, jolloin ne on voitu sijoittaa kunkin yhteydentarjoajan tiloihin ja pienentää verkon alueellisista toimintahäiriöistä johtuvia haittoja.

- Yleispalvelut

  Verkon avaintoiminnot, eli yhteydentarjoajien valinta, DHCP-palvelut, tukiasemien RADIUS-viestien käsittely ja nimipalvelu on keskitetty palvelinpariin, joka ei kuitenkaan sisällä varsinaisia kahdennusmekanismeja. Näitä palveluita kutsutaan yleispalveluiksi.

- Nimipalvelin

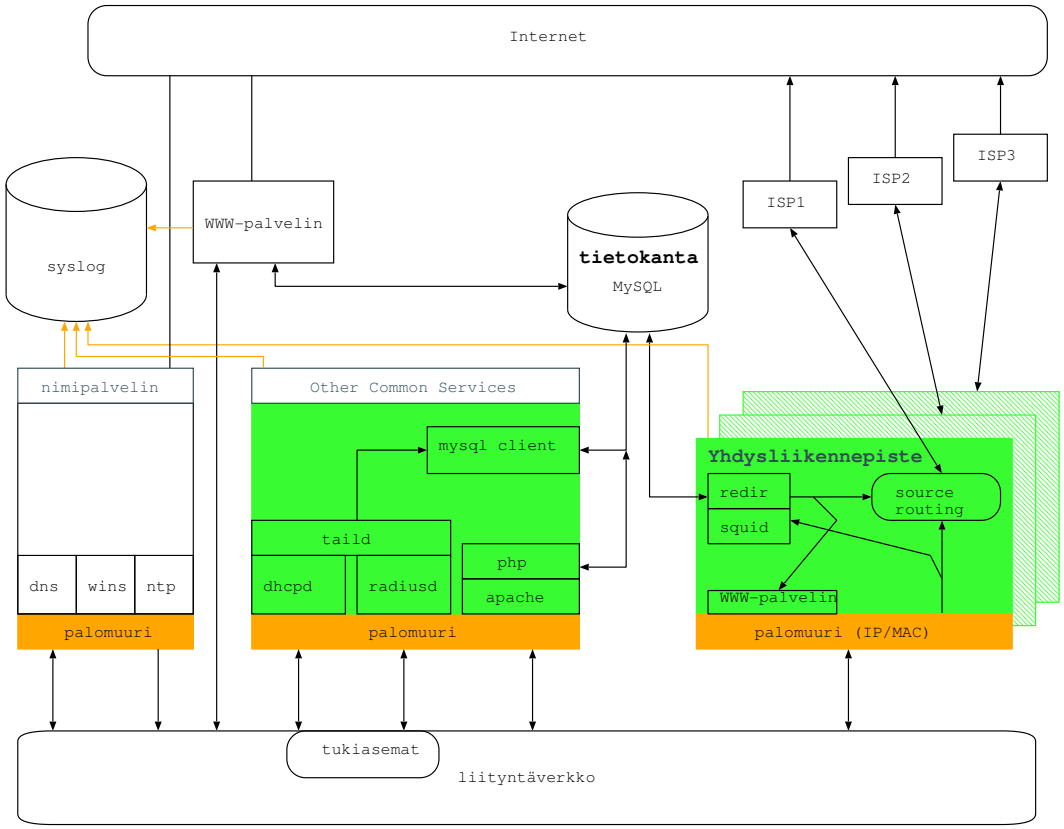
  Nimi- eli DNS-palveluun sisältyy luonnostaan hajautus ensi- ja toissijaisiin palvelimiin. WLPR.NET-verkossa sisäinen nimipalvelu on laitteistosyistä toteutettu yleispalvelukoneiden avulla eikä erillisillä nimipalvelinkoneilla.

- Tietokanta

  Verkon MySQL-tietokanta, jossa säilytetään mm. DHCP-palvelusta ja verkon tukiasemilta saatua tietoa, on asennettu omaan erilliseen palvelinkoneeseensa.

3.2.1 Yhdysliikennepiste

Yhdysliikennepiste toimii verkon käyttäjien oletusyhdyskäytävänä, ja kaikki liikenne verkon käyttäjien ja Internetin välillä kulkee sen kautta. Yhdysliikennepisteestä voidaan käyttää myös englanninkielistä nimeä *Access Controller*. Verkossa voi olla yksi tai useampia yhdysliikennepisteitä.



Kuva 4: Lappeenranta-malli

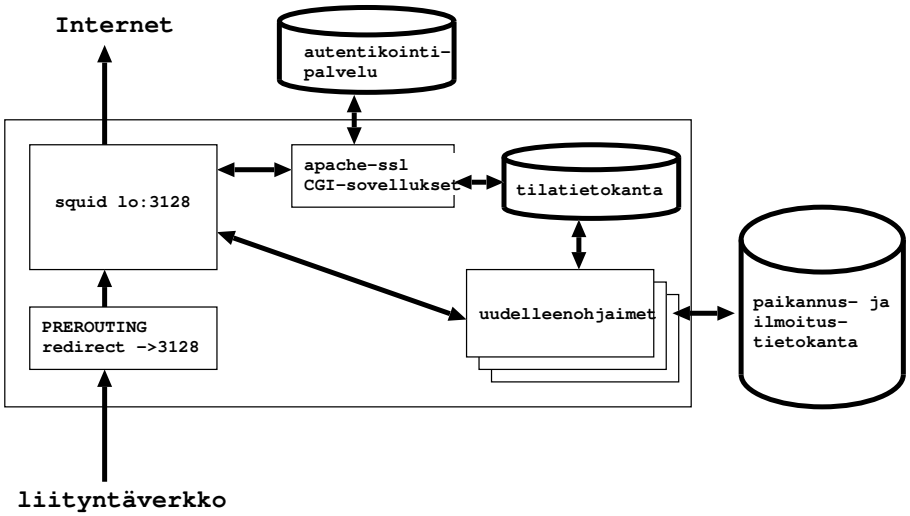
Yhdysliikennepiste suorittaa pakettien reititystä lähdeosoitteen perusteella mikäli se palvelee useampaa kuin yhtä palveluntarjoajaa. Pakettien lähdeosoitteeseen perustuva reititys voidaan toteuttaa esimerkiksi Linuxin iproute2-työkaluilla, jolloin jokaista yhteydentarjoajaa varten on oma reititystaulunsa.

Yhdysliikennepiste voi toimia myös läpinäkyvänä WWW-välityspalvelimena, joka kykenee näyttämään käyttäjille tiedotteita ja mainoksia Lappeenranta-mallin määrittelemällä tavalla perustuen käyttäjien fyysiseen sijaintiin. Käyttäjät joilla ei ole sopimusta Internet-yhteydentarjoajan kanssa voivat käyttää erikseen määriteltyjä paikallisia palveluita.

3.2.2 Yhdysliikennepisteen esimerkkitoteutus

WLPR.NET-verkon ja Lappeenrannan teknillisen yliopiston tarpeisiin luotiin tämän diplomityön osana yhdysliikennepiste, jonka ominaisuuksiin kuuluvat käyttäjien RADIUS-autentikointi, IP- ja MAC- osoitepohjainen suodatus, läpinäkyvä WWW-välimuisti mainosominaisuuksilla, nimipalvelu sekä käyttäjien automaattinen uloskirjaus. Yhdysliikennepisteen rakenne on esitetty kuvassa 5. Yhdysliikennepisteen lähdekoodi on saatavissa WLPR.NET -verkon kotisivuilta.

Läpinäkyvä WWW-välimuisti on toteutettu Linuxin iptables-työkalujen REDIRECT-ominaisuudella, joka ohjaa kaiken tcp-porttiin 80 suuntautuvan liikenteen paikalliseen porttiin, jossa squid -välimuistipalvelin kuuntelee. Mikäli WWW-välimuisti halutaan sijoittaa erilliseen laitteeseen tai laiteryppääseen, voidaan liikenteen ohjaus toteuttaa Linuxissa merkitsemällä halutut tcp-paketit iptablesin merkintäominaisuudella ja reitittämällä paketit oikeaan paikkaan iproute2-työkalulla.



Kuva 5: Yhdysliikennepisteen esimerkkitoteutus

Yhdysliikennepisteen WWW-välimuistipalvelimeen on liitetty uudelleenohjaimia (engl. *redirector*), joiden tehtävänä on tarkistaa, täytyykö käyttäjälle näyttää

jokin muu WWW-sivu kuin mitä hän alun perin haki. Näitä sivuja ovat kehotus käyttää DHCP:tä, mainokset, ilmoitukset ja sisäänkirjautumissivu. Sekä DHCP-kehotus, mainossivut että ilmoitukset perityvät Lappeenranta-mallista.

Sisäänkirjautumisen onnistuessa käyttäjää varten luodaan liikenteen reititys- ja laskentasäännöt. Lisäksi sisäänkirjautumisaika, MAC- ja IP-osoitteet sekä käyttäjätunnus merkitään paikalliseen BerkeleyDB-tietokantaan. Yhdysliikennepisteen väärinkäyttöä toiseksi käyttäjäksi tekeytymällä vaikeuttaa Linuxin iptables-työkaluilla tehty palomuuri, joka tarkistaa että käyttäjän MAC- osoite vastaa sitä IP-osoitetta jonka käyttäjä on saanut DHCP-palvelimelta. [15] Sisäänkirjautumissivu tuotetaan paikallisella CGI-ohjelmistolla, joka toimii apache-ssl -palvelimen alaisuudessa. CGI-ohjelmisto on toteutettu perl-kielellä, ja se sisältää liitynnät ulkoiseen RADIUS-tietokantaan sekä sisäiseen BerkeleyDB-tietokantaan, jossa säilytetään tilatietoa sisäänkirjautuneista käyttäjistä.

Valvontaprosessi *watchdog* on toteutettu perl-kielellä ja liitetty samaan BerkeleyDB-tietokantaan muiden järjestelmän osien kanssa. *Watchdog* valvoo sisäänkirjautuneiden käyttäjien tilaa lähettämällä näiden IP-osoitteisiin “ICMP echo request” -paketteja. Mikäli käyttäjän laite vastaa ping-pakettiin, kirjataan vastauksen aika paikalliseen BerkeleyDB-tietokantaan. Ellei vastausta kuulu määriteltävään aikaan mennessä, kirjataan käyttäjä automaattisesti ulos järjestelmästä. Ylläpidon helpottamiseksi järjestelmään kuuluvat myös tilatietoa tarjoava WWW-sovellus sekä komentorivisovellukset käyttäjien uloskirjaamiseen ja tietokannan puhdistamiseen vikatilanteiden varalta. Myös nämä sovellukset on toteutettu perl-kielellä. Järjestelmän valvontaa varten järjestelmän eri osat tuottavat lokiviestejä Unixin syslog-rajapinnan kautta.

3.2.3 Yleispalvelut

Yleispalvelukoneiden avulla tuotetaan Lappeenranta-mallin toiminnalle oleelliset erityispalvelut, jotka ovat yhteisiä kaikkien palveluntarjoajien kesken. Näiden palvelujen tarjoamiseen tarvitaan eräitä ohjelmistokomponentteja:

- WWW-palvelin tarjoaa käyttäjille sivun, jonka avulla nämä voivat valita yhteydentarjoajan. Yhteydentarjoajan valinta perustuu käyttäjän laitteiston MAC-osoitteeseen.
- DHCP-palvelin antaa verkon käyttäjille IP-osoitteita, jotka kuuluvat ennalta määriteltyihin yhteydentarjoajakohtaisiin osoiteavaruuksiin. Jokaista yhteydentarjoajaa kohti on oma osoiteavaruutensa.
- RADIUS-palvelin vastaanottaa verkon tukiasemien lähettämiä viestejä, joiden perusteella verkon käyttäjät voidaan paikantaa tukiasemakohtaisesti. Tämä edellyttää tukiasemilta toimivaa RADIUS-tukea.

Kaikki nämä komponentit on yhdistetty keskitettyyn tietokantaan, johon tallennetaan tiedot asiakkaiden saamista IP-osoitteista, yhteydentarjoajan valinnasta sekä fyysisestä sijainnista verkossa.

3.2.4 Nimipalvelu

Nimipalvelun tehtävänä on nimensä mukaisesti tuottaa nimi- ja muita oheispalveluita, jotka eivät ole verkon toiminnan kannalta kriittisiä mutta kuitenkin erittäin tärkeitä. DNS (Domain Name Server)-palvelin on konfiguroitu toimimaan ensisijaisena nimipalvelimena liityntäverkon palveluille. WLPR.NET-verkon tapauk-

sessa tällaisia palveluita ovat esimerkiksi www.wlpr.net, backup.wlpr.net ja gamesrv.wlpr.net, ja palveltava toimialue on wlpr.net. Toimialuetiedostojen siirto on sallittu yhdysliikennepisteissä toimiville toissijaisille nimipalvelimille. Näin verkon käyttäjät voivat selvittää sekä paikallisen verkon että Internetin DNS-nimiä.

3.2.5 Tietokantapalvelin

Tietokantapalvelin sisältää MySQL-relaatiotietokannan. MySQL-tietokanta hyväksyy tcp-yhteyksiä erikseen määrittellyiltä asiakkailta, kuten verkon yleispalveluja tuottavilta koneilta ja verkon yhdysliikennepisteiltä. Tietokannan rakenne on esitetty taulukossa 3.

Tietokantaan on tallennettu tiedot esimerkiksi verkon tukiasemista, mainoksista, käyttäjien sijainnista, käyttäjämääristä ja usein kysytyistä kysymyksistä vastauksineen.

Taulukko 3: Tietokannan rakenne

Taulu	*Sisältö*
accesspoint	Verkon tukiasemat: IP-osoite, nimi, tyyppi, ohjelmistoversio
advertisement	Verkon käyttäjille näytettävät ilmoitukset
area	Paikannustietojen alueet
diary	Ylläpidon päiväkirjamerkinnät
faq	Usein kysytyt kysymykset
group_aps	Tukiasemien ryhmät
last_adv	Kullekin käytäjälle viimeksi näytetty ilmoitus
leases	Käyttäjien IP-osoitetiedot DHCP-palvelimen lokitiedostosta
location	Käyttäjien viimeisin paikannustieto: MAC-osoite, tukiaseman IP-osoite
mac_id	MAC-osoitteiden valmistajakohtaiset etuliitteet
organizations	Verkkoon liittyvät organisaatiot
services	Verkon palvelulista
show_ads	Näytetäänkö ilmoituksia
spot	Pistemäisiä paikkatietoja
stat_ap	Käyttäjien paikannustiedot statistiikkaa varten
stat_date	Tukiasemakohtaiset käyttäjätilastot

3.3 Lappeenranta-mallin oheispalvelut WLPR.NET-verkossa

WLPR.NET -verkkoa varten Lappeenranta-mallia on laajennettu eräillä komponenteilla, jotka parantavat verkon käytettävyyttä ja hallittavuutta sekä käyttäjän että ylläpidon näkökulmasta. Näitä komponentteja ovat:

- WWW-palvelin

  WWW-palvelimelle (www.wlpr.net) on tallennettu julkiset sivut, joilta käyttäjät saavat tietoa verkosta ja sen käytöstä. Palvelin on kytketty sekä liityntäverkkoon että Internet-verkkoon, joten siihen tallennetut tiedot ovat saatavilla molempia teitä. Näin verkosta kiinnostuneet voivat tutustua verkon käyttöohjeisiin ja muuhun materiaaliin ennen langattoman laitteiston hankintaa. WWW-palvelimessa sijaitsevat myös verkon hallintaan ja valvontaan keskittyvä sivukokonaisuutensa admin.wlpr.net, sekä projektin dokumentointiin käytettävä project.wlpr.net.

- Loki- ja varmuuskopiointipalvelin Lokipalvelin kerää lokitietoa verkon muista palvelimista ja tallentaa tiedot keskitetysti. Tiedon analysointia varten käytetään erityisiä työkaluja. Palvelinta käytetään myös verkon muiden komponenttien varmuuskopiointiin. Pääsy palvelimeen on rajattu. Palvelu toteutetaan *syslog-ng*-ohjelmistolla.

- Julkinen nimipalvelin Julkinen nimipalvelin huolehtii verkon domainnimeen liittyvien palvelujen nimistä sekä sähköpostin reitityksestä. Nämä toiminnot toteutetaan *ISC Bind 9* ja *Postfix* -ohjelmistoilla.

4 LAPPEERANTA-MALLIN ARVIOINTIA

Operaattoririippumattoman verkon rakentajan tulee voida perustella verkon arkkitehtuurin valintaa sekä teknisillä että taloudellisilla syillä. Tässä kappaleessa arvioidaan Lappeenranta-mallin ominaisuuksia ja niistä syntyvää kokonaisuutta verkon rakentajan näkökulmasta, jotta voitaisiin vastata kysymyksiin, kuten *Miksi operaattoririippumattoman verkon toteutustavaksi tulisi valita Lappeenranta-malli?* ja *Miten Lappeenranta-malli soveltuu erilaisiin käyttökohteisiin ja kuinka suuria käyttäjämääriä se voi palvella?*

4.1 Lappeenranta-mallin ominaisuuksia

Ominaisuuksiltaan Lappeenranta-malli on ainutlaatuinen, erityisesti paikannus- ja mainostusominaisuuksiensa vuoksi. Mallin soveltuvuutta verkon ja sen palveluiden rakentamiseen voidaan kuitenkin arvioida useista eri näkökulmista, joista ohessa muutamia.

4.1.1 Kustannukset

Lappeenranta-mallin käyttöönottokustannukset ovat suhteellisen pienet. Tarvittavat ohjelmistot ovat vapaasti saatavilla, joten kustannukset muodostuvat laitteistohankinnoista ja työstä. Laitteiston hinnassa voidaan päästä suuriin säästöihin, jos luotettavuudesta tingitään ja käytetään esimerkiksi muusta käytöstä jo poistuneita PC-tietokoneita. Langattoman verkon rakennuskustannukset riippuvat suuresti kohdealueen ominaisuuksista ja halutusta signaalinvoimakkuudesta. Laitteistokustannukset pienenevät jatkuvasti laitteiden halventuessa, mutta toisaalta tukiasemien sijoitus ja runkoyhteydet voivat olla kalliita-

kin. Langattoman lähiverkon kustannuksia pinta-alan funktiona on siten mahdotonta arvioida yleispätevästi.

4.1.2 Skaalautuvuus

Lappeenranta-mallin skaalautuvuutta parantaa mahdollisuus mallin komponenttien hajauttamiseen useisiin tietokoneisiin. Näin suorituskyvyn ongelmia voidaan tarvittaessa ratkaista laitteita lisäämällä. Teoriassa Lappeenranta-malli skaalautuu helposti tuhansien käyttäjien verkkoihin, mutta näin suuriin käyttäjämääriin ei päästäne julkisissa alueverkoissa vielä muutamaan vuoteen, ellei langaton lähiverkkotekniikka yleisty matkapuhelimissa.

4.1.3 Käyttövarmuus ja tietoturva

Verkon käyttövarmuutta tulee arvioida sitä palvelutasoa vasten, jonka verkon palveluntarjoaja on määritellyt tavoitteekseen ja toisaalta luvannut asiakkailleen. Internet-yhteydet ovat muuttumassa yhä jokapäiväisemmäksi asiaksi, itsestäänselvyydeksi, jota ihminen tarvitsee puhelimen tavoin päivittäiseen asiointiin. Siksi Internet-yhteyksien käyttövarmuuden tavoitetasoksi tulee asettaa vertailukelpoisuus puhelinverkon käyttövarmuuteen.

Puhelinverkon käyttövarmuuteen liittyviä seikkoja on määritelty myös lainsäädännössä. Telemarkkinalaki sekä erinäiset liikenneministeriön ja Telehallintokeskuksen määräykset muodostavat perustan teleyritysten toiminnalle. Tämän vuoksi on selvää, että Internet-yhteyksien tarjonnassa verkon ylläpito talkootyönä voi olla toimiva malli vain äärimmäisen harvoissa tapauksissa.

Verkon käyttövarmuutta voidaan parantaa useilla keinoilla, joista tärkeimmät ovat:

- Hyvä suunnittelu

  Hyvällä suunnittelulla voidaan vaikuttaa merkittävästi rakennettavan verkon käyttövarmuuteen. Suunnitelmien tulee aina olla kirjallisia. Suunnitelmista muodostuu verkon dokumentaatio, jota tulee päivittää aina kun verkkoon tehdään muutoksia. Verkon topologia, laitteiston fyysinen sijoittelu, kaapelointi, laitteiden elinkaari ja huoltosopimukset tulisi määritellä käyttövarmuutta silmälläpitäen.

- Riskien kartoitus

  Verkon toimintaan liittyvien riskien kartoitus kannattaa aina. Kun tiedetään, mitkä riskit ovat, voidaan niiden aiheuttamiin ongelmiin varautua ja minimoida verkon toiminnalle koituvat haitat. Riskianalyysiä tehtäessä tulee huomioida:

  - henkilöuhat sekä sisältä- että ulkoapäin
  - verkkkoon murtautumiset
  - tulipalot,vesivahingot,fyysiset murrot ja ilkivalta
  - sota,terrorismi ja luonnon katastrofit
  - muut uhat

- Laitteiston käyttövarmuuden parantaminen

  Laitteistoksi tulisi hyväksyä vain vikasietoisilla ja ajon aikana vaihdettavilla komponenteilla (levyjärjestelmä, virtalähteet) varustettuja tietokoneita. Verkon laitteille kannattaa tehdä huoltosopimukset, joilla voidaan taata haluttu palvelutaso.

- Toiminnan valvonta

Verkon toiminnan kattava valvonta on erityisen tärkeää nopean vikatilanteista toipumisen ja toisaalta myös vikatilanteiden ennakoinnin kannalta. Verkon valvonnalla voidaan vastata myös moniin tietoturvaan liittyviin kysymyksiin. Valvonnalla ei silti saa vaarantaa verkon käyttäjien yksityisyyden suojaa.

4.1.4 Ylläpidettävyys

Ylläpidettävyydeltään Lappeenranta-malli edustaa tyypillistä vapaista ohjelmistokomponenteista yhdistelemällä rakennettua tietojärjestelmää. Järjestelmän ylläpito edellyttää näiden komponenttien hyvää tuntemusta, mutta toisaalta osaava henkilö voi hallita ja räätälöidä järjestelmää varsin tehokkaasti. Eräitä toimintoja helpottamaan on laadittu WWW-sovelluksia. Vapaiden ohjelmistokomponenttien mukanaan tuoma avoimuus luo loistavat mahdollisuudet verkon turvalliseen ja keskitettyyn etähallintaan. Tällä tavoin pieni ylläpitoryhmä voi hallita keskitetysti useita erillisiä alueverkkoja.

4.1.5 Siirrettävyys ja käyttöönotto

Lappeenranta-malli on teknisesti täysin siirrettävissä käytettäväksi muiden verkkojen luonnissa toisille paikkakunnille. Avoimet ohjelmistokomponentit mahdollistavat vapaan räätälöinnin kunkin verkon erityistarpeisiin. Verkon pystytys vaatii osaavat henkilöresurssit, sillä Lappeenranta-mallin verkkokomponenttien tuotteistus on yleisesti katsoen matalalla tasolla.

Lappeenranta-malli on sellaisenaan valmis otettavaksi käyttöön avoimessa julkisessa alueverkossa. Avoin arkkitehtuuri tarjoaa rajattomat mahdollisuudet pai-

kalliseen räätälöintiin, jolloin mallia tai osia siitä voidaan hyödyntää mitä erilaisimmissa verkkoympäristöissä. Mallin käyttöönotto vaatii tietoliikennetekniikan, TCP/IP:n ja Unixin perusteiden hyvää osaamista. Koska avoimen julkisen alueverkon ylläpito on vaativa tehtävä, on käyttöönottoon tarvittavien erityistaitojen merkitys lopulta melko vähäinen. Paljon tärkeämpää on verkon tarkka dokumentointi muotoon, jota on helppo hallita ja josta verkon ylläpitäjät voivat helposti omaksua tarvittavat tiedot.

Jokaisen malliin kuuluvan ohjelmistokomponentin tulisi olla selkeästi lisensoitu, jotta organisaatio tai yksityinen henkilö voi varmistua oikeudestaan ohjelmiston käyttöön, muuttamiseen ja edelleen levittämiseen. Lappeenranta-mallin lisenssinä on ohjelmistojen vapauden varmistava GNU GPL. Tietoverkkojen perusosina toimii nykyisin lähes pelkästään vapaita ohjelmistokomponentteja (kuten ISC:n DHCPd ja BIND, Linux, Apache), joten on eduksi, että Lappeenranta-mallin komponentit seuraavat samoja suuntaviivoja mahdollisimman selkeästi.

4.2 Lappeenranta-mallin testaus

Lappeenranta-mallin ominaisuuksien tutkimiseksi järjestelmälle suoritettiin joukko testejä, joiden tavoitteena oli selvittää arkkitehtuurin suorituskykyä ja toimintavarmuutta. Järjestelmän suorituskyvyn kannalta olennaisin komponentti on HTTP-välityspalvelin, jonka kautta kaikki käyttäjien WWW-liikenne ohjataan. Tämän vuoksi testausta painotettiin HTTP-välityspalvelimen suorituskyvyn mittaamiseen.

Web Polygraph on WWW-välimuistipalvelimien suorituskyvyn mittaamiseen tarkoitettu vapaasti saatavilla oleva ohjelmisto. [3] Web-polygraph pyrkii simuloimaan todellista http-liikennettä standardoitujen testikuormien avulla.

Lappeenranta-mallin version 2.0 WWW-välimuistin kapasiteetin selvittämiseksi rakennettiin testiympäristö, jossa suorituskykyä voitiin tutkia simuloimalla todellisen julkisen verkon HTTP-liikennettä.

HTTP-välityspalvelimena toimi Lappeenranta-mallin mukaisesti Squid. Squid on vapaaseen lähdekoodiin perustuva laajalle levinnyt välityspalvelinohjelmisto, joka toimii useilla eri laitteisto- ja käyttöjärjestelmäalustoilla.

4.2.1 Testilaitteisto ja -verkko

Testit suoritettiin Lappeenrannan teknillisen yliopiston tietotekniikan laitoksella. Testiympäristöön koostui seuraavista tietokoneista ja tietoliikennelaitteista:

- Yhdysliikennepiste
  CPU: AMD Athlon XP 2200+, 512MB RAM
  IDE-levy: 40GB Maxtor 6Y060L0 (7200rpm, 9ms, ATA-133, 2MB cache)
  SCSI-levy: 18GB IBM DNES-318350W (7200rpm, 7ms, U2 LVD, 2MB cache)
  SCSI-ohjain: Adaptec 19160B
  Verkkokortit: 3Com 3c905C, 3Com 3C996B-T, integroitu BCM95702A20

- Polygraph-asiakas
  CPU: AMD Athlon XP 1700+, 512MB RAM
  IDE-levy: 40GB IC35L040AVER07-0 Verkkokorti: 3Com 3c905C

- Polygraph-palvelin
  CPU: Pentium II 400Mhz, 128MB RAM
  IDE-levy: 20GB IBM-DTLA-305020 Verkkokortti: 3Com 3c905C

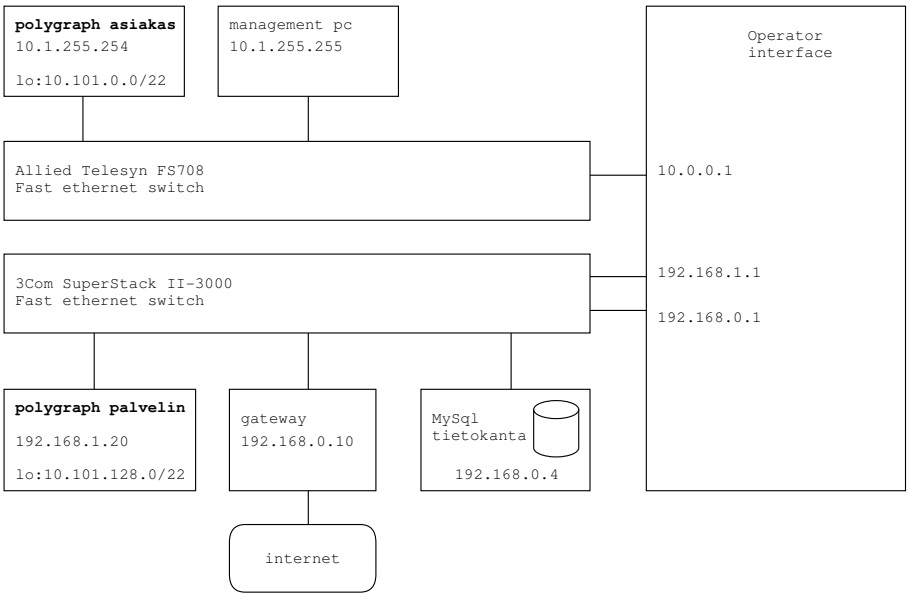
- Tietokantapalvelin
  CPU: Pentium III 1Ghz, 256MB RAM
  IDE-levy: 40GB IC35L040AVER07-0 Verkokortti: 3Com 3c905C

- Ethernet-kytkin: 3com Superstack II 3000, 24x 10/100 porttia

  Ethernet-kytkin: Allied Telesyn FS708, 8x 10/100 porttia

Laitteista rakennettiin testiverkko, jonka rakenne on esitetty kuvassa 6. Testiverkon toimivuus varmistettiin *netperf*-ohjelmistolla. Välityskyvyksi saatiin noin 81 Mbit/s asiakkaalta palvelimelle ja noin 87 Mbit/s palvelimelta asiakkaalle päin. 2 Asiakkaassa ja palvelimessa käytetyillä verkkokorteilla olisi mahdollista päästä 92-95Mbit/s tasolle [4]. Alhaisempi nopeus johtunee pääosin käytetystä ensimmäisen sukupolven Fast-ethernet kytkimestä (3Com Superstack II 3000). Tämän nopeuden alentumisen merkitystä testien lopputulokseen voidaan kuitenkin pitää varsin vähäisenä, sillä testikuorman tarvitsema kaistanleveys oli vain noin 10 megabittiä sekunnissa.

Verkon yhdysliikennepiste oli konfiguroitu reitittämään liikennettä Polygraph-testiosoiteavaruuksien 10.101.0.0/22 ja 10.101.128.0/22 välillä. Yhdysliikennepiste toimi läpinäkyvänä välityspalvelimena, eli se poimi tcp-porttiin 80 kohdistuneet yhteydet ja ohjasi ne Squidille, jonka konfiguraatio on esitetty liitteessä 4.



Kuva 6: Testiverkon rakenne

Web Polygraph konfiguroitiin käyttämään testikuormaa Polymix-4. Testikuorman konfiguraatio on esitetty liitteessä 3.

4.2.2 Ohjelmistot

Lähes kaikissa testiverkon tietokoneissa käytettiin Debian GNU/Linux -käyttöjärjestelmän versiota 3.0r1. Linux-käyttöjärjestelmän asetuksia oli välttämätöntä muuttaa testejä varten. Asetukset on kuvattu liitteessä 1. HTTP-välityspalvelimena toimi Squid-2.5STABLE1, jonka käännösoptiot ja asetukset on esitetty liitteessä 4. HTTP-välityspalvelimen suorituskykyä mitattiin Web Polygraph -ohjelmistolla.

Testiaineistona käytettiin polymix-4 -koekuormaa. Web Polygraphin polymix-4 -testikuorman asetukset on esitetty liitteessä 3. Polygraph-palvelimessa käytettiin Measurement Factoryn paketoimaa ja optimoimaa FreeBSD 4.3 -käyttöjärjestelmää. [3] Verkon toiminta ja tietoliikenteen välityskyky varmistettiin netperf -ohjelmistolla. Tulokset ovat liitteessä 2

4.2.3 Testitulokset

Testien tarkoituksena oli selvittää miten tietokannan ja uudelleenohjainprosessien käyttö vaikuttaa squidin suorituskykyyn. Oli odotettavissa, että suorituskyky heikkenee, sillä tietokantakyselyiden teko vaatii aina aikaa. Lisäksi perl-kielellä kirjoitettua uudelleenohjainta tai tietokantaa ei oltu optimoitu suorituskykyä ajatellen. Alkuperäinen tarkoitus oli suorittaa varsinainen testaus käyttäen apuna Linux-rypästä, jossa on kahdeksan tehokasta PC-tietokonetta liitettynä toisiinsa Gigabit-ethernet -verkolla. Tämä kävi kuitenkin tarpeettomaksi kun huomattiin,

että riittävän suuri testikuorma voidaan käsitellä kahdella tietokoneella joista toinen toimii asiakkaana ja toinen palvelimena.

Testiympäristön rakentaminen oli monimutkaista. Testijärjestelmä ei alussa ollut kovinkaan vakaa, mikä johtui väärin valituista asetuksista sekä testialustassa että testien kohteessa. Useita Linux-käyttöjärjestelmän oletusasetuksia piti muuttaa. Muutokset on kuvattu liitteessä 1.

Squidin ylikuormittuessa HTTP–pyyntöjä alkoi jäädä jonoon kaikkien uudelleenohjainten ollessa aktiivisena. Tällaisessa tilanteessa HTTP–pyyntöjono saattoi kasvaa hallitsemattomasti mikä johti squidin virhetilanteeseen ja uudelleenkäynnistymiseen, joka edelleen hidasti järjestelmän toimintaa. Tämän ongelman estämiseksi squid konfiguroitiin 'redirector_bypass on' -optiolla, jolloin jonossa olevat HTTP-pyynnöt ohittavat uudelleenohjaimet mikäli niitä ei ole yhtään vapaana. Käytännössä tästä seuraa, että HTTP-välityspalvelimen ylikuormittuessa käyttäjä ei ehkä näe mainosta tai ilmoitusta jonka hän olisi muutoin nähnyt.

Uudelleenohjaimen merkityksen selvittämiseksi testejä tehtiin sekä uudelleenohjaimen kanssa että ilman sitä. Laitteiston vaikutusta vasteaikoihin tutkittiin käyttämällä välimuistin tallennusvälineenä sekä IDE- että SCSI-kiintolevyjä. Vasteajat kaikissa tilanteissa on esitetty graafisesti kuvissa 7, 8 ja 9. Testien yhteenveto on esitetty taulukossa 4.

Testeissä kävi ilmi, että squidin kyky välittää HTTP-liikennettä ei olennaisesti muutu uudelleenohjaimen suorittamien tietokantahakujen vuoksi. Tämä tulee ilmi erityisesti käytettäessä SCSI-levyä, jolloin laitteistosta aiheutuvat viiveet ovat pienet. Parhaiten ero on näkyvissä kuvassa 9, jossa vasteajat uudelleenohjaimen kanssa ja ilman sitä seuraavat tarkasti toisiaan. Yksittäisen HTTP-pyynnön ja vastauksen välinen viive kasvaa jopa yli 200 millisekunnilla, kun

välimuisti on IDE-levyllä. SCSI:n toimintatapa kuormittaa välimuistina toimivan tietokoneen prosessoria huomattavasti IDE-väylää vähemmän, ja niinpä SCSI-levyä käytettäessä uudelleenohjaimen aiheuttamat viiveet jäävät pienemmiksi: erot ovat vain muutamien kymmenien millisekuntien luokkaa.

Vasteaikojen ero IDE:n ja SCSI:n välillä on selvästi suurempi, kuin uudelleenohjaimen aiheuttama viive. Uudelleenohjain ei siis vaikuta squidin välityskykyyn, vaan ainoastaan hakujen vasteaikoihin. Koska vaikutus vasteaikoihin on pienehkö, voidaan Vladislav Kurzin diplomityössään esittämää ratkaisua [9] pitää käyttökelpoisena.

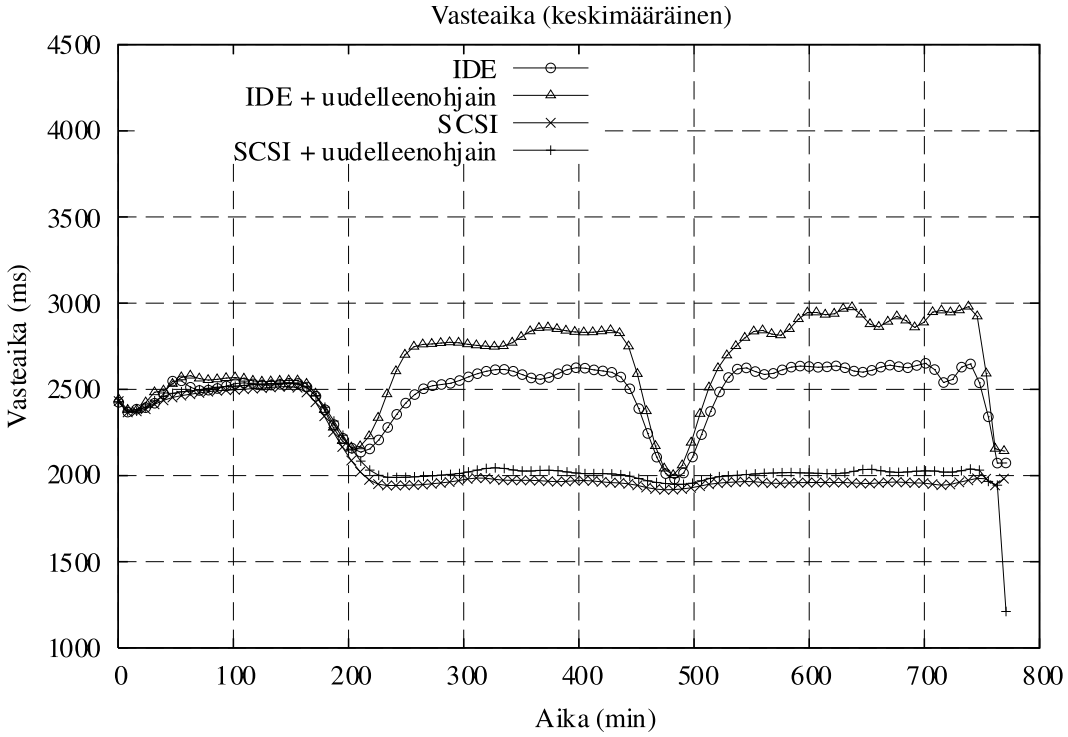
	IDE		SCSI	
	uudelleenohjain	*ei uudelleenohjainta*	*uudelleenohjain*	*ei uudelleenohjainta*
Välityskyky	150.36 rep/sec	150.44 rep/sec	150.43 rep/sec	150.43 rep/sec
Kaistanleveys	8.27 Mbit/s	8.25 Mbit/s	8.35 Mbit/s	7.87 Mbit/s
Vasteaika	2883.30 msec	2619.51 msec	2018.57 msec	1959.32 msec
- *ohitukset*	3638.50 msec	3320.52 msec	2632.92 msec	2592.92 msec
- *osumat*	797.83 msec	570.82 msec	157.08 msec	123.63 msec
Osumatarkkuus	29.57%	28.24%	26.51%	27.36%
Virheet	0.00%	0.00%	0.00%	0.00%

Taulukko 4: Polymix-4 -testin tulokset testin vaiheessa *top2*.

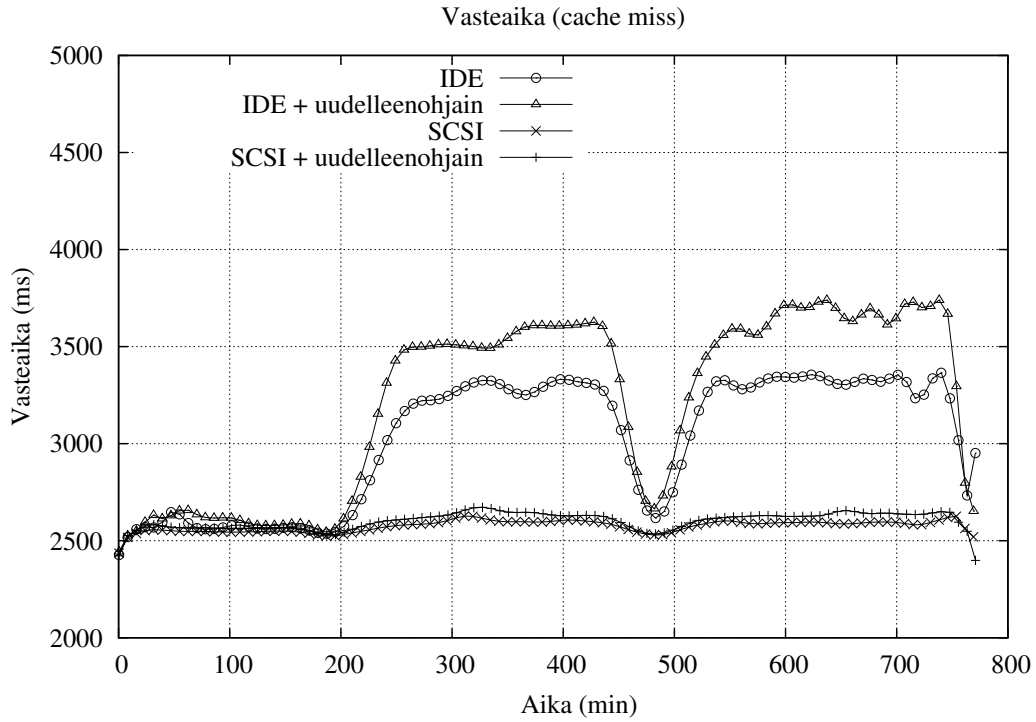
4.2.4 Tulosten analysointia

Squidin, kuten muidenkin HTTP-välimuistipalvelimien suorituskyky on pitkälti riippuvainen palvelimen levyjärjestelmän suorituskyvystä [1]. Testattu laitteisto oli tyypillinen pc-työasema, jonka levyjärjestelmän suorituskyky on korkeintaan välttävä. Tästä syystä oli perusteltua käyttää pientä, vain yhden gigatavun kokoista levytallennustilaa.

Mikäli uudeleenohjaimen vaikutusta vasteaikoihin halutaan pienentää, tulisi ratkaisua hakea ensisijaisesti tietokannan toiminnasta. Tietokantahaut ja tietokannan indeksointi pitäisi optimoida paremmin. Mikäli tällä ei päästäisi

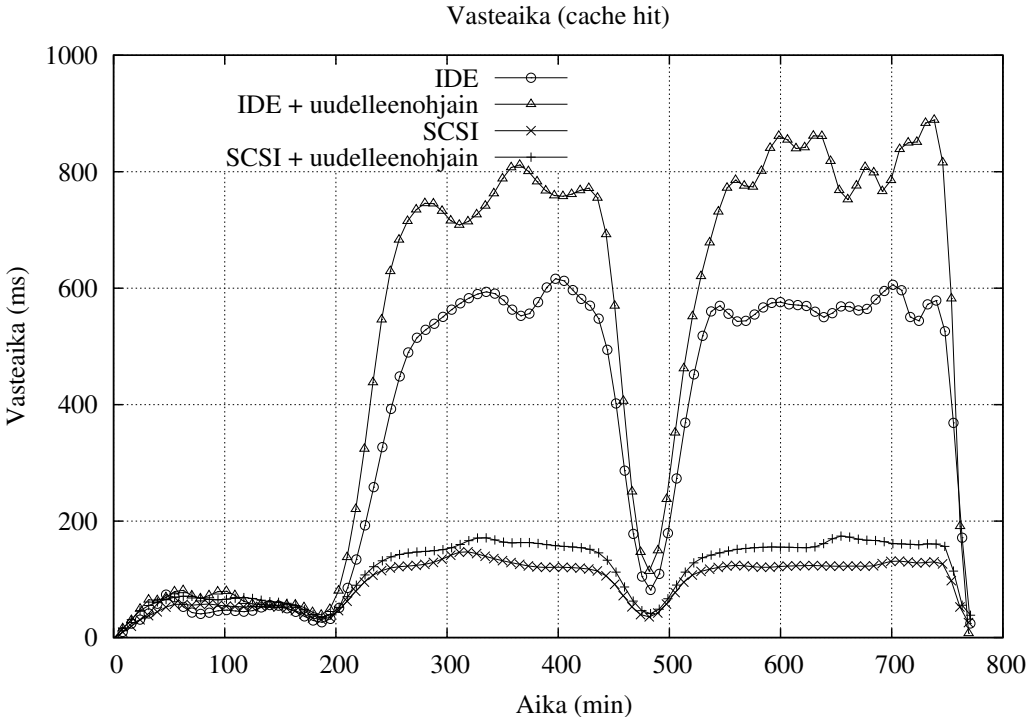


Kuva 7: Keskimääräiset vasteajat



Kuva 8: Vasteajat välimuistin ohi

riittävään tulokseen, voitaisiin suorituskykyä hakea kirjoittamalla uudelleenohjain Perlin sijasta C-kielellä. Suurempiin parannuksiin vasteajoissa, var-



Kuva 9: Vasteajat välimuistista

sinkin haun osuessa välimuistissa olevaan dokumenttiin, päästäisiin nopeutamalla välimuistipalvelimen levyjärjestelmää. Squidin suorituskykyä rajoittavaksi tekijäksi nousi testeissä Squid-alustana toimineen pc-koneen hidas IDE-levyjärjestelmä. Yksittäisellä, vaikkakin modernilla IDE-levyllä varustettu PC ei sovellu HTTP-välimuistiksi raskaaseen käyttöön. Vakavampaan käyttöön on siis syytä harkita yhdellä tai useammalla SCSI-levyllä varustettua laitteistoa. Testatun kaltainen laitteisto riittää kuitenkin hyvin palvelemaan pieniä ja keskisuuria verkkoja, joiden yhtäaikainen käyttäjämäärä ei kohoa muutamaa kymmentä suuremmaksi.

Merkittävä osa tämänpäivän interaktiivisesta verkkoliikenteestä on HTTP-liikennettä. Sen vuoksi http-välityspalvelin on keskeisessä roolissa koko järjestelmän suorituskykyä arvioitaessa. IP-pakettien reititykseen tarvittavat laitteistoresurssit ovat pienet tiettyyn rajaan saakka (noin 100Mbit/s).

5 JOHTOPÄÄTÖKSET

5.1 Alueverkkojen tarve

Yhteiskunnasta on tulossa tietoliikenneyhteyksien infrastruktuurin rakentaja. Tämän perusrakenteen päällä kaupalliset tahot voivat kehitellä palveluita. Analogia toimii esimerkiksi tieverkkoon: valtio tekee tiet (myös syrjäseuduille) ja yksityiset kuljettavat. Langattomat lähiverkot täydentävät langallisia yhteyksiä ja tarjoavat liikkuvuutta niille käyttäjille jotka sitä tarvitsevat. Langattomilla lähiverkoilla ei kuitenkaan vielä lähivuosina voi olla merkittävää roolia hajaasutusalueiden verkottamisessa, sillä verkkojen kantamat ovat pieniä. Yhteiskunnan apu on välttämätöntä mikäli halutaan varmistaa alueellinen tasa-arvo laajakaistaisten yhteyksien saatavuudessa.

5.2 Verkon toimintamallin valinta

Lappeenranta-malli tarjoaa hyvät edellytykset kokonaan tai osittain mainosrahoitteisen operaattoririippumattoman verkon luomiseen. Paikkatiedon hyväksikäyttö mainonnassa lisännee verkon kiinnostavuutta mainostajien silmissä. Toisaalta langattomien lähiverkkojen tämänhetkiset vaatimattomat käyttäjämäärät tekevät niistä marginaalisen mainosmedian. Operaattoririippumattoman verkon toteutuksen suurimmat käytännön ongelmat liittyvät WLPR.NET-koeverkossa saatujen kokemusten perusteella sopimusteknisiin seikkoihin, eivät niinkään teknisiin ratkaisuihin. Monioperaattoriympäristön toimintaa tukee Lappeenranta-mallin tarjoama mahdollisuus toteuttaa tehokas tiedotuskanava verkon ylläpitäjiltä käyttäjille.

Operaattoririippumattoman verkon rakentajan täytyy siis joko valita jokin valmis malli autentikointijärjestelmän pohjaksi tai kehittää kokonaan oma järjestelmä. Eräät mallit ovat jo saavuttaneet suosiota alkuperäisen käyttökohteensa ulkopuolella, ja onkin perusteltua kysyä sopisivatko ne Lappeenranta-mallia paremmin operaattoririippumattoman verkon toteuttamiseen. Kysymykseen voidaan etsiä vastausta vertailemalla eri mallien ominaisuuksia.

5.3 Lappeenranta-mallin käyttökokemukset

Lappeenranta-malliin perustuva yhdysliikennepiste otettiin julkiseen käyttöön Lappeenrannan teknillisessä yliopistossa vuonna 2003, jonka jälkeen se on palvellut tavallisia verkon käyttäjiä melko luotettavasti jo seitsemän vuoden ajan: Ongelmia ja lyhyitä palvelukatkoja ovat aiheuttaneet lähinnä BerkeleyDB -tietokannan vanhempien versioiden ohjelmistovirheet. Laitteiston suunnittelussa otettiin huomioon tämän diplomityön suorituskykytestien tulokset, minkä ansiosta yhdysliikennepisteen suorituskyky on osoittautunut käytännössä riittäväksi, vaikka laitteisto onkin jo vanhentunut. Riittävästä suorituskyvystään huolimatta yhdysliikennepiste on lopulta tullut elinkaarensa päähän hallinnollisten syiden vuoksi. [1]

Lappeenranta-malli kehittyy ja jatkaa toimintaansa muiden yhdysliikennepisteiden voimin. Näissä yhdysliikennepisteissä varhaisemman, tässä diplomityössä esitellyn yhdysliikennepisteen esimerkkitoteutuksen ominaisuuksia on laajennettu IP-liikenteen profiloinnilla, jonotuskäytännöillä, viestinnän laadun takaamiseen tähtäävillä toimenpiteillä (Quality of Service, QoS) sekä kertakäyttöisiin käyttäjätunnus-salasanapareihin perustuvalla sisäänkirjautumisjärjestelmällä. Näillä keinoin vähäinenkin tiedonsiirtokapasiteetti on saatu riittämään lukuisille yhtäaikaisille käyttäjille. Virtualisointiteknologian voimakkaan kehityksen ansios-

ta fyysisiä palvelinkoneita on voitu korvata kevyemmillä ja toimintavarmemmilla virtualisoiduilla ratkaisuilla.

[1]Lappeenrannan teknillinen yliopisto korvaa Lappeenranta-malliin perustuvan yhdysliikennepisteen vuonna 2010. Tilalle asennetaan kaupallinen tuote.

LÄHTEET

[1] Glenn Chisholm. Squid performance as a factor of the number of disks utilised. Tekninen raportti, http://www.squid-cache.org/Benchmarking/Number-of-Disks/index.html, 2003.

[2] International Engineering Consortium. Wireless local loop (wll). Tekninen raportti, IEC, http://www.iec.org/online/tutorials/wll/, 2003.

[3] Measurement Factory. *Web Polygraph Technical documentation*. http://www.web-polygraph.org, 2003.

[4] Netperf group. Public netperf database. Tekninen raportti, Netperf group, http://www.netperf.org/netperf/numbers/NetperfBrowse.html, 2003.

[5] Liikenne ja viestintäministeriö. Käyttäjä verkossa: Kunnat ja laajakaistaiset liityntäyhteydet. Kirjassa *Liikenne- ja viestintäministeriön julkaisuja*, osa 10. Liikenne- ja viestintäministeriö, 2003.

[6] Liikenne ja viestintäministeriö. Ruotsin ja suomen laajakaistayhteyksien kattavuus. Kirjassa *Liikenne- ja viestintäministeriön julkaisuja*, osa 21. Liikenne- ja viestintäministeriö, 2003.

[7] Liikenne ja viestintäministeriö. Valokaapeli suomen runko- ja alueverkoissa 2002. Kirjassa *Liikenne- ja viestintäministeriön julkaisuja*, osa 20. Liikenne- ja viestintäministeriö, 2003.

[8] Matti Juutilainen. Yhdysliikennepisteen suunnittelu. Diplomityö, Lappeenrannan teknillinen korkeakoulu, 2001.

[9] Vladislav Kurz. Delivery system for location based information in wireless ip networks. Diplomityö, Lappeenrannan teknillinen korkeakoulu, 2002.

[10] Viestintäministeri Suvi Lindén. puhe Laajakaista kaikille -hankkeen aluesemінaarissa 2.4.2009, huhtikuu 2009.

[11] Andrew Malis ja William Simpson. RFC 2615: PPP over SONET/SDH, kesäkuu 1999.

[12] NoCat.net. *NoCat technical documentation.* NoCat.net, http://nocat.net, 2002.

[13] Hedenfalk M. Pehrson B. Pelletta E., Lilieblad F. The design and implementation of an operator neutral open wireless access network at the kista it-university. Kirjassa *Proceedings of the 12th IEEE Workshop on Local and Metropolitan Area Networks, Stockholm-Kista*, 2002.

[14] Sotatalousosasto. Sotatekninen arvio ja ennuste (stae 2005). Tekninen raportti, Puolustusvoimat, 2003.

[15] Radek Spáčil. Forcing usage rules in public wireless lans. Diplomityö, Lappeenrannan teknillinen korkeakoulu, 2002.

[16] William Stallings. *Data and Computer Communications.* Prentice-Hall, 1997.

Liite 1: Linux-käyttöjärjestelmään tehdyt muutokset

Linux-käyttöjärjestelmää piti muuttaa, jotta sekä squid että polygraph saattoivat suoriutua testistä. 64k prosessikohtaisen tiedostokuvaajan sallivat muutokset tehtiin sekä Operator Interfaceen että Polygraph-asiakkaaseen.

```
--- linux-2.4.20/include/linux/posix_types.h    Thu Jan 20 20:48:35 2000
+++ linux-2.4.20.patched/include/linux/posix_types.h    Fri Apr  4 17:03:44 2003
@@ -22,7 +22,7 @@
 #define __NFDBITS      (8 * sizeof(unsigned long))

 #undef __FD_SETSIZE
-#define __FD_SETSIZE   1024
+#define __FD_SETSIZE   65536

 #undef __FDSET_LONGS
 #define __FDSET_LONGS  (__FD_SETSIZE/__NFDBITS)


--- /usr/include/bits/types.h.orig      Mon Apr  7 10:24:37 2003
+++ /usr/include/bits/types.h   Fri Apr  4 17:04:15 2003
@@ -95,7 +95,7 @@


 /* Number of descriptors that can fit in an ‘fd_set’.  */
-#define __FD_SETSIZE   1024
+#define __FD_SETSIZE   65536


 typedef int __key_t;
```

Operator interface

Jotta järjestelmän resurssit saatiin riittämään testien läpäisyyn oli tarpeen muuttaa eräitä asetuksia.

Koska testaus luo paljon yhtäaikaisia tcp-yhteyksiä, piti ip_conntrack-moduulin kapasiteettia kasvattaa lisäämäällä rivi /etc/sysctl.conf -tiedostoon:

```
net/ipv4/ip_conntrack_max=65536
```

Squidin levyvälimuisti sijaitsi partitiolla /dev/hda8. Partitio mountattiin eli liitettiin osaksi hakemistopuuta optiolla “noatime”, jolloin tiedostojen viimeisintä käsittelyaikaa ei päivitetä tiedostoja luettaessa. Tämä nopeuttaa toimintaa, ja Squid huolehtii itse kirjanpidosta.

```
/dev/hda8 on /squid type ext3 (rw,noatime)
```

Polygraph-asiakas

Polygraph-asiakkaassa piti asetaa Linuxin tcp-pinon ominaisuus “tcp_rfc1337” käyttöön. Asetus ratkaisee ongelmat, jotka on esitetty RFC1337:ssa (TIME-WAIT Assassination Hazards in TCP).

```
debian:~# echo "1" > /proc/sys/net/ipv4/tcp_rfc1337
```

Liite 2: Testiverkon suorituskyky mitattuna netperf-ohjelmistolla

```
server# netperf -l 30 -H 192.168.1.20 -t TCP_STREAM
TCP STREAM TEST to 192.168.1.20
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 16384  16384  16384    30.00      81.66

server# netperf -l 30 -H 192.168.1.20 -t TCP_STREAM
TCP STREAM TEST to 192.168.1.20
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 16384  16384  16384    30.00      81.14

client# netperf -l 30 -H 10.1.255.254 -t TCP_STREAM
TCP STREAM TEST to 10.1.255.254
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    30.00      87.33

client# netperf -l 30 -H 10.1.255.254 -t TCP_STREAM
TCP STREAM TEST to 10.1.255.254
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    30.00      87.38
```

Liite 3: Polymix-4 testiaineiston määrittely

Polymix-4 -testiaineisto määriteltiin seuraavasti:

```
#include "/usr/local/polygraph/workloads/benches.pg"
Bench TheBench = benchPolyMix4;
TheBench.client_side.addr_space = [ 'lo::10.101.0-3.1-250/22' ];
TheBench.server_side.addr_space = [ 'lo::10.101.128-131.1-250:80/22'
    ];
TheBench.client_side.hosts = [ '10.1.255.254' ];
TheBench.server_side.hosts = [ '192.168.1.20' ];
TheBench.peak_req_rate = 150/sec;
rate FillRate = 75% * TheBench.peak_req_rate;
size ProxyCacheSize = 1GB;
DnsResolver Resolver;
#include "/usr/local/polygraph/workloads/polymix-4-guts.pg"
```

Liite 4: Squid -WWW-välimuistipalvelimen käännösoptiot ja asetukset

Squid -palvelinohjelmisto käännettiin seuraavilla valitsimilla:

```
./configure --sysconfdir=/etc/squid --localstatedir=/var/local/
    squid
--enable-linux-netfilter --enable-async-io '--enable-removal-
    policies=lru heap'
```

Squidin käynnistyksessä tarkistettiin, että prosessi saa käyttöönsä suurimman mahdollisen määrän tiedostokuvaajia

```
# ulimit -HSn 65563
```

Squid -palvelinohjelmiston ajonaikainen konfiguraatio oli seuraava:

```
### WLPR.NET squid cache configuration

### We usually do not need to use ICP with upstream caches.
icp_port 0

### We use async-io (maybe) to get more throughput
cache_mem 256 MB
cache_dir aufs /squid/cache 1000 24 256

# some tuning settings
cache_swap_low 95
maximum_object_size 32 MB
maximum_object_size_in_memory 6 KB
ipcache_size 2048
fqdncache_size 2048
cache_replacement_policy lru
memory_replacement_policy heap GDSF

### This is the program for delivering advertisements, auth &
    whitelist
redirect_program /usr/local/sbin/ads.pl
redirect_children 50
redirector_bypass on
#       This is used to pass parameters to the various
    authentication
#       schemes.
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
```

```
#Suggested default:
#refresh_pattern ^ftp:           1440    20%     10080
#refresh_pattern ^gopher:        1440    0%      1440
#refresh_pattern .               0       20%     4320
#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny to_localhost

# Exampe rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
acl our_networks src 10.0.0.0/8 192.168.0.0/16
http_access allow our_networks

# And finally deny all other access to this proxy
http_access deny all

# Allow replies to client requests. This is complementary to
    http_access.
http_reply_access allow all

# Allow ICP queries from everyone
icp_access allow all

### Debian has already exiting system user for us.
cache_effective_user proxy
cache_effective_group proxy

### This is the preferred hostname for proxy
visible_hostname proxy.wlpr.net

### We run in transparent proxy mode
httpd_accel_host virtual
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
### We provide some anonymity to our users
forwarded_for off
# Leave coredumps in the first cache dir
coredump_dir /squid/cache
```

```
# logging
cache_access_log none
cache_log       /var/log/squid/cache_log
cache_store_log none
```

Liite 5: Yhdysliikennepisteen esimerkkitoteutuksen lähdekoodi

Kirjastofunktiot

```
# Access controller libraries for WLPR.NET / (C) Tomi Lapinlampi

# This program is free software; you can redistribute it and/or
    modify
# it under the terms of the GNU General Public License as published
    by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.

package access_controller;
require Exporter;

use BerkeleyDB ;
use MLDBM qw(BerkeleyDB::Hash) ;
use Unix::Syslog qw(:subs :macros);
use Authen::Radius;
use CGI qw(-no_xhtml :all :cgi-lib :standard);
use POSIX qw(setsid);

our @ISA    = qw(Exporter);
our @EXPORT_OK = qw($env);
our @EXPORT = qw(readconfig %config open_env open_db $env logout
    wdhms
                 radiuslogin iptables db_insert check_bytes
                     check_arp
                 pageheader daemonize);


###
### Config file handling
###

sub readconfig {

    my $config_file="/etc/access_controller.conf";

    open (CONFIG,"$config_file") || die "cannot open $config_file";
    while (<CONFIG>) {
        chomp;                  # no newline
        s/#.*//;                # no comments
        s/^\s+//;               # no leading white
        s/\s+$//;               # no trailing white
        next unless length;     # anything left?
        my ($var, $value) = split(/\s*=\s*/, $_, 2);
        $config{$var} = $value;
    }
```

```
    close (CONFIG);
}

###
### BerkeleyDB functions
###

sub open_env {
    my $env = new BerkeleyDB::Env(
                                    -Flags=>DB_INIT_MPOOL|DB_CREATE,
                                    -Home=> $_[0],
                                    );
    die "Could not create env: $! ".$BerkeleyDB::Error. "\n" if !
       $env;
    return $env;
}

sub open_db {
    my( $file, $Rhash, $env ) = @_;
    my $db_key = tie( %{$Rhash}, 'MLDBM',
                      -Flags=>DB_CREATE,
                      -Filename=>$file,
                      -Env=>$env );
    die "Can't open $file: $! ".$BerkeleyDB::Error."\n" if !$db_key;
    return $db_key;
}

###
### User logout: ($mac,$ip_addr,$user,$db,$ads_db)
###

sub logout
{
    ($mac,$ip_addr,$user,$db,$ads_db,$public_iface)=@_;
    # clean up database
    $db->db_del($mac);
    $db->db_sync;
    $ads_db->db_del($ip_addr);
    $ads_db->db_sync;
    ($result,$reason)=iptables($mac,$ip_addr,"-D");
    return ($result,$reason);

}

sub radiuslogin {
    # returns 1 if success
    my ($user,$password,$mac,$ip_addr,$db,$ads_db)=@_;
    $r = new Authen::Radius(Host => $config{radius_server},
                                    Secret => $config{
                                        radius_password});
    my $auth_result=$r->check_pwd($user,$password);
    return $auth_result;

}

sub db_insert {

    my ($user,$mac,$ip_addr)=@_;
```

```
    my $login_time=time;
    $env = open_env($config{dbdir});
    $db = open_db( "acl_state", \%acl_state, $env );
    $ads_db = open_db( "ads_table",\%ads_table, $env);
    $acl_state{$mac}= [$user,$ip_addr,$login_time,$login_time];
    $ads_table{$ip_addr}=$user;
    $db->db_sync();
    $ads_db->db_sync();

}


sub iptables {

    my ($mac,$ip_addr,$function)=@_;
    $position="";
    if ($function eq "-I"){
        $position=1;
    }

    my @cmd;
    $cmd[0]="$config{iptables} -t nat $function POSTROUTING
        $position \
            -o $config{public_iface} -s $ip_addr/32 -j SNAT \
            --to $config{public_ip}";
    $cmd[1]="$config{iptables} $function fwd_chain $position \
            -s $ip_addr -d 0/0 -m mac --mac-source $mac -j ACCEPT";
    $cmd[2]="$config{iptables} $function fwd_chain $position \
            -d $ip_addr -s 0/0 -j ACCEPT";
    $cmd[3]="$config{iptables} -t filter $function squid_chain \
            -s $ip_addr -j RETURN";
    $cmd[4]="$config{iptables} -t filter $function squid_chain \
            -d $ip_addr -j RETURN";

    foreach $command (@cmd) {
        system ($command);
    }

    return (0,"none");

}


sub wdhms {
    my( $weeks,$days,$hours,$minutes,$seconds,$sign,$res ) = qw/0 0
        0 0 0/;

    $seconds = shift;
    $sign    = $seconds == abs $seconds ? '' : '-';
    $seconds = abs $seconds;

    ($seconds, $minutes) = ($seconds % 60, int($seconds / 60)) if
        $seconds;
    ($minutes, $hours  ) = ($minutes % 60, int($minutes / 60)) if
        $minutes;
    ($hours,   $days   ) = ($hours   % 24, int($hours   / 24)) if
        $hours;
    ($days,    $weeks  ) = ($days    %  7, int($days    /  7)) if
```

```
        $days;

    $res = sprintf '%d seconds',     $seconds;
    $res = sprintf "%d minutes $res", $minutes if $minutes or $hours
        \
    or $days or $weeks;
    $res = sprintf "%d hours $res", $hours if $hours or $days or
        $weeks;
    $res = sprintf "%d days $res", $days if $days or $weeks;
    $res = sprintf "%d weeks $res", $weeks if $weeks;

    return "$sign$res";
}

sub check_bytes {
    my $ip_addr=shift(@_);
    open (VALUES,"$config{iptables} -L fwd_chain -nvx|") || \
    die "cannot open iptables";
    while (<VALUES>){
        next unless (/.*$ip_addr.*/);
        ($pkts,$counter,$rest,$rest,$rest,$rest,$rest,$src,$rest)=
            split;
        if ($src =~ /^0.*/ ) {
            $routedbytes[0]=$counter;
        }
        if ($src =~ /^10.*/)
        {
            $routedbytes[1]=$counter;
        }
    }
    close (VALUES);

    open (VALUES,"$config{iptables} -L squid_chain -nvx|") \
    || die "cannot open iptables";
    while (<VALUES>){
        next unless (/.*$ip_addr.*/);
        ($pkts,$counter,$rest,$rest,$rest,$rest,$rest,$src,$rest)=
            split;
        if ($src =~ /^0.*/ ) {
            $squidbytes[0]=$counter;
        }
        if ($src =~ /^10.*/)
        {
            $squidbytes[1]=$counter;
        }
    }
    close (VALUES);

    $bytes[0]=$routedbytes[0]+$squidbytes[0];
    $bytes[1]=$routedbytes[1]+$squidbytes[1];

    return @bytes;
}


sub check_arp {
    my $ip_addr=shift(@_);
    my ($ip_from_arp,$type,$flags,$mac,$mask,$device);
```

```
    open (ARP,"</proc/net/arp");
    while (<ARP>){
        next if (/^IP/); # strip headers
        ($ip_from_arp,$type,$flags,$mac,$mask,$device)=split;
        last if ($ip_from_arp eq $ip_addr);
    }
    return $mac;
}



sub pageheader {

    my $pageident = shift(@_);

    my $title = "$config{pagetitle}: $pageident";

    print
        start_html(-title=>$title,
                   -author=>'',
                   -base=>'true',
                   -BGCOLOR=>'white',
                   -LINK=>'#A10337',
                   -VLINK=>'#808080'),
        center(
               p,
               img {src=>'/lty2.gif',
               alt=>'Lappeenranta University of Technology'},
               img {src=>'/wlpr_small.png',
               alt=>'Wireless Lappeenranta: WLPR.NET'},
               p

               );
}

sub daemonize {
    chdir '/'                 or die "Can't chdir to /: $!";
    open STDIN, '/dev/null'   or die "Can't read /dev/null: $!";
    defined(my $pid = fork)   or die "Can't fork: $!";
    exit if $pid;
    setsid                    or die "Can't start a new session: $
        !";
    umask 0;
}
1;
```

Kirjautumistoiminnon toteuttava ohjelma *login.pl*.

```
#!/usr/bin/perl -w

# Login script for WLPR.NET / (C) Tomi Lapinlampi
# requires libauthen-radius-perl in Debian

# This program is free software; you can redistribute it and/or
    modify
```

```
# it under the terms of the GNU General Public License as published
    by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.

use lib "/usr/local/sbin";
use Authen::Radius;
use CGI qw(-no_xhtml :all :cgi-lib :standard);
use Unix::Syslog qw(:subs :macros);
use BerkeleyDB ;
use MLDBM qw(BerkeleyDB::Hash) ;
use access_controller;

use vars qw( %h $k $v ) ;

readconfig();

# settings
my $debug=0;
umask 0111; # :-)


my $log_ident='Internet access';
my $SYSLOG_LEVEL = "auth.info";
my ($FACILITY, $PRIORITY);
($FACILITY = $SYSLOG_LEVEL) =~ s/(\w+)\.(\w+)/LOG_\U$1/;
($PRIORITY = $SYSLOG_LEVEL) =~ s/(\w+)\.(\w+)/LOG_\U$2/;
my $iptables = "/usr/local/sbin/iptables" ;

# end of settings

# Get the web environment
ReadParse();

my $user=$in{user};
my $password=$in{password};
my $ip_addr=$ENV{'REMOTE_ADDR'};
my $now = time;

$env = open_env($config{dbdir});
$db = open_db( "acl_state", \%acl_state, $env );
$ads_db = open_db( "ads_table",\%ads_table, $env);

print header(-type  =>  'text/html',
             -expires  =>  'now');

openlog("$log_ident", LOG_PID, eval "$FACILITY");

pageheader("Login");

# fetch the MAC of the client
$mac = check_arp($ip_addr);

if ( $acl_state{$mac} )     # user is currently logged in
{
    if ( $in{Logoff} ){

        # user is logging out, handle it
```

```
    @bytes=check_bytes($ip_addr);
    $user=$ads_table{$ip_addr};
    $session_start=$acl_state{$mac}[2];

    print center("Logging out $user...");

    ($result,$reason)=logout($mac,$ip_addr,$user,$db,$ads_db,\
                             $config{public_iface});

    if ($result == 0){

        &logmsg ("Logout OK from $ip_addr ($mac) by $user");

        print center(
                    "Logout complete",
                    br,
                    "You transferred a total of ",
                    $bytes[0] + $bytes[1],
                    " bytes during this session.",
                    br,
                    "Your total session time was ",
                    wdhms($now - $session_start),
                    br,
                    "Use",
                    a({-href=>'https://login.wlpr.net'},
                    "https://login.wlpr.net"),
                    " to log in again.",br
                    );
    } else {
        print center("Something went wrong. Contact somebody.");
        print center ("Reason: $reason");

    }

    &closedb;
    print end_html;
    exit;
}

# user is logged in, show some statistics

print center(
            p,"You're already logged in.",br,
            "Your MAC address is $mac"
            );

@bytes=check_bytes("$ip_addr");
$totalbytes=$bytes[0] + $bytes[1];

print center(

             br,"You have transferred $totalbytes bytes",
             "($bytes[0] in, $bytes[1] out) during this session
                 .",
             start_form(-method=>'POST',
                        -action=>'https://login.wlpr.net/'),
             hidden('Logoff','1'),
```

```
                    p,submit(-name=>'Logoff')
                    );
} else {

# user is not logged in yet, show login page

    if ($in{user}){
        print center(
                    "Authenticating...",br
                   );
        &authenticate($in{user},$in{password});
        &closedb;
        print end_html;
        exit;
    } else {
        # user needs to log in
        &loginpage;
    }
}

print end_html;
&closedb;
exit;


sub loginpage {
    print
        center(
               p,
               start_form(-method=>'POST',
                          -action=>'https://login.wlpr.net/'),
               "Login: ",
               textfield(-name=>'user',
                         -size=>8,
                         -maxlenght=>8),
               p,"Password: ",
               password_field(-name=>'password',
                              -size=>8,
                              -maxlenght=>8),
               hidden('url',$in{url}),
               p,submit(-name=>'Login'),
               p,"Use your LUT unix account to sign in.",
               p,a({-href=>'http://www.wlpr.net'},"Back to WLPR\.NET
                   ")
               );
}



sub authenticate {

    if (radiuslogin($user,$password) == 1){

        iptables($mac,$ip_addr,"-I");
        db_insert($user,$mac,$ip_addr);

        &logmsg ("Login OK from $ip_addr ($mac) by $user");
```

```
        print center(
                    "Authentication succeeded for $user",br,
                    "You have now access to the Internet",br
                    );

        if ($in{url}) {
            print center(
                        "You may now continue browsing to ",
                        a({-href=>$in{url}},"$in{url}")
                        );
        }

        print center(
                    p,
                    "You will be logged out automatically after 1",
                    "hour of inactivity (no respond to ICMP ping)",
                        br,
                    "You can also come back to ",
                    a({-href=>'https://login.wlpr.net'},"this page
                        "),
                    " to log out."
                    );


    } else
    {
        &logmsg ("Login FAILED from $ip_addr ($mac) by $user");
        print center(
                    "Authentication failed",br,
                    a({-href=>'https://login.wlpr.net'},"Try again
                        ")
                    );

    }

    closelog;
}



sub logmsg
{
    syslog(eval "$PRIORITY", "%s", "@_");
}

sub closedb {
    $db->db_close(); ### NEW
    $ads_db->db_close(); ### NEW

}
```

Käyttäjien istuntoja valvova *watchdog.pl*.

```
#!/usr/bin/perl -w
#
# Access controller watchdog for WLPR.NET / (C) Tomi Lapinlampi
```

```
# This program is free software; you can redistribute it and/or
    modify
# it under the terms of the GNU General Public License as published
    by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.

use lib "/usr/local/sbin";
use Unix::Syslog qw(:subs :macros);
use Net::Ping;
use BerkeleyDB ;
use MLDBM qw(BerkeleyDB::Hash) ;
use POSIX qw(setsid);
use access_controller;

#my $debug=1;
#my $sleepTime = 20;
my (%acl_state, %ads_table, $v, $reason);

readconfig();
$env = open_env($config{dbdir});

###
### Syslog settings
###
$log_ident='watchdog';
$SYSLOG_LEVEL = "daemon.info";
my ($FACILITY, $PRIORITY);
($FACILITY = $SYSLOG_LEVEL) =~ s/(\w+)\.(\w+)/LOG_\U$1/;
($PRIORITY = $SYSLOG_LEVEL) =~ s/(\w+)\.(\w+)/LOG_\U$2/;
sub logmsg {
    if ($debug){
        print "@_\n"; return;
    }
    syslog(eval "$PRIORITY", "%s", "@_");

 }

openlog("$log_ident", LOG_PID, eval "$FACILITY");
&logmsg ("$log_ident starting up... ");

daemonize or die "cannot fork" unless $debug;

while (1)
{

    $db = open_db( "acl_state", \%acl_state, $env );
    $ads_db = open_db( "ads_table", \%ads_table, $env );

    foreach $mac (sort keys %acl_state)
    {

        $nrofusers=scalar(keys %acl_state);
        &logmsg ("monitoring $nrofusers users");

        $ping = Net::Ping->new("icmp");
        $user = $acl_state{$mac}[0];
```

```
        $ip_addr = $acl_state{$mac}[1];

        &logmsg ("Pinging $mac: $acl_state{$mac}[1]") if $debug;

        if ($ping->ping($acl_state{$mac}[1], $config{ping_timeout}))
        {
            &logmsg ("Host $acl_state{$mac}[1] was alive, updating
                database")\
                 if $debug;
            @temp=@{$acl_state{$mac}};
            $db->db_del($mac) ;
            $acl_state{$mac}=[$temp[0],$temp[1],$temp[2],time];
            $db->db_sync();
        } else
        {
            &logmsg ("Host $acl_state{$mac}[1] was down") if $debug;
        }

        &logmsg ("Closing ping for $mac") if $debug;
        $ping->close();

        $now=time;

        $idletime=$now-$acl_state{$mac}[3];
        logmsg ("$user has been idle for $idletime seconds") if
            $debug;


        if (($now-$config{max_idle_time}) > $acl_state{$mac}[3]){

            ($result,$reason)=logout($mac,$ip_addr,$user,$db,\
                $ads_db,$config{public_iface});

            if ($result == 0){
                &logmsg ("Logging out ($user $ip_addr, $mac ),
                    reason: PING timeout");
            }
            else {
                &logmsg ("ERROR Logging out ($user $ip_addr, $mac )
                    ");
            }
        }
        sleep $config{ping_interval};

    }
    $db->db_close();
    $ads_db->db_close();
    sleep $config{ping_interval};
}

closelog;
```